Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Ciência da Computação

# Controle de Fluxo Adaptativo para Gateways Bluetooth Low-Energy Aplicado a Sistemas de Monitoramento Remoto de Pacientes

Danilo Freire de Souza Santos

Tese de Doutorado submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Engenharia de Software

Hyggo Oliveira de Almeida, D.Sc
Angelo Perkusich, D.Sc
(Orientadores)

Campina Grande, Paraíba, Brasil

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG**

S237c Santos, Danilo Freire de Souza.

Controle de fluxo adaptativo para Gateways Bluetooth Low-Energy aplicado a sistemas de monitoramento remoto de pacientes / Danilo Freire de Souza Santos. – Campina Grande, 2016.

137 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2016.

"Orientação: Prof. D.Sc. Hyggo Oliveira de Almeida, Prof. D.Sc. Angelo Perkusich".

Referências.

1. Redes Sem Fio. 2. Internet. 3. Bluetooth. 4. Controle de Fluxo. 5. Qualidade de Serviço. I. Almeida, Hyggo Oliveira de. II. Perkusich, Angelo. III. Título.

CDU 004.771(043)

# Resumo

O cenário introduzido pela Internet das Coisas potencializa a criação de um novo conjunto de aplicações e serviços, onde diversos dispositivos interagem entre si através da Internet. Esse cenário viabiliza o advento de novas tecnologias de transmissão sem fio de baixo consumo, como o Bluetooth Low-Energy (BLE), as quais viabilizam a criação de redes pessoais (PAN) sem fio. Em paralelo, com a crescente disponibilidade de Dispositivos Pessoais de Saúde com capacidade de comunicação, um cenário onde informações de saúde podem ser disponibilizadas na Internet surge, viabilizando a criação de sistemas de Saúde Conectada. Entretanto, ao viabilizar a criação de redes PAN interconectando diversos dispositivos, a Qualidade de Serviço na rede necessária para o correto funcionamento desses dispositivos de saúde pode ser afetada, dado que redes PAN BLE não oferecem mecanismos para a diferenciação da Qualidade de Serviço entre os dispositivos conectados. Considerando esse contexto de compartilhamento de uma rede PAN entre diversos dispositivos, nesse trabalho propõe-se uma abordagem para o controle do fluxo adaptativo em Gateways BLE utilizando um mecanismo de distribuição de créditos temporal entre os clientes de uma rede PAN. Para essa priorização, informações fornecidas por aplicações são utilizadas para a distribuição e configuração dos parâmetros de conectividade dos dispositivos da rede PAN. Também são apresentados detalhes sobre o projeto e evolução arquitetural do controlador adaptativo, detalhando suas características de controle de fluxo com prioridade temporal. São apresentados resultados experimentais do funcionamento do controlador adaptativo em diferentes cenários. Esses resultados demonstram que o mesmo é capaz de garantir a Qualidade de Serviço de rede necessária para dispositivos específicos em um ambiente compartilhado. Para a validação desse trabalho em um cenário mais amplo, é apresentada uma arquitetura para Sistemas de Monitoramento Remoto de Pacientes padronizado para a Internet das Coisas. Esse sistema serve como base para a implantação e avaliação experimental do controlador de fluxo adaptativo em um *Smart-Gateways* BLE, onde informações de serviços e aplicações em saúde são utilizadas para priorizar Dispositivos Pessoais de Saúde a depender do seu contexto de uso.

# Abstract

The Internet of Things paradigm enables a new set of applications and services to be available in the Internet. This scenario makes possible the development of new low-power communication technologies, such as Bluetooth Low-Energy (BLE), which creates wireless Personal Area Networks (PAN). At the same time, the rising availability of Personal Health Devices (PHD) capable of PAN communication and the desire of keeping a high quality of live are the ingredients of the Connected Health vision. However, as the number of PHDs increase, the number of other peripherals connected in the PAN also increases. Therefore, PHDs are now competing for medium access with other devices, decreasing the network Quality of Service of health applications in the BLE PAN, as these networks do not guarantee Quality of Service requirements for connected devices. In this context, where a BLE PAN is shared with multiple devices, it is where this work is immersed. In this work is presented an approach for adaptive flow-control of BLE Gateways using a temporal credit distribution mechanism between clients in a PAN. For this distribution, application context information is used for network prioritization and parameter configuration of PAN devices. In this work is detailed how the adaptive flow-control was designed and how was its architectural evolution, detailing how its temporal mechanism works. Experimental results are presented showing the controller behavior in different scenarios. These results show that using the proposed approach it is possible to guarantee Quality of Service requirements for target devices using a prioritization process in a shared medium. In order to validate this work in a broad scenario, it is also presented a standard-based Remote Patient Monitoring System architecture for the Internet of Things. This system is used as base infrastructure for prioritization of PHDs connections based on their state and requirements by the use of a Smart BLE Gateway. An implementation was developed showing the relevance of the problem and how a BLE adaptive controller can assist in the prioritization of devices in the context of healthcare services and applications.

# Agradecimentos

Em primeiro lugar, agradeço a minha família por todo apoio, compreensão e suporte durante minha vida.

Também agradeço a minha companheira Estela por todo apoio e paciência de conviver comigo durante esses anos (do mestrado pra cá, não foi pouco). Não deve ter sido fácil, mas todos saíram vivos e felizes.

Nesses tempos modernos (e para poupar papel), não posso esquecer-me de agradecer aos meus amigos aqui representados por grupos no *whatsapp*: grupo *AAmigos* de antigos companheiros (as) de cerveja; grupo *CPFA2* e suas discussões de sexta a noite; grupo *Lorota* e a cervejinha paulistana; grupo *Cervejinha* de velhos amigos; grupo *LowLevel* por manter o alto nível de conteúdo; e o grupo *FreeNatal* que sempre bate a porta no final de ano. Como adendo aos agradecimentos, não posso esquecer-me dos novos sobrinhos e sobrinhas. Todos me ajudaram a desopilar um pouco (pelo jeito virei tiozão mesmo).

Agradeço aos professores e orientadores Angelo Perkusich e Hyggo Almeida pelas discussões, dicas, paciência e direcionamentos durante o desenvolvimento desse trabalho. Também agradeço aos professores da banca avaliadora, Kyller Gorgônio, Leandro Dias, Marcos Morais e Augusto Venâncio Neto pelas orientações e criticas para o direcionamento desse trabalho.

Não posso esquecer-me dos colegas contemporâneos de doutorado na COPIN e na COPELE pelas discussões sobre os temas de trabalhos, prazos, e estresses compartilhados (e comemorações de qualificações e teses). Também agradeço as equipes da Signove e do laboratório Embedded por todo suporte durante esse doutorado.

Agradeço também a CAPES pelo financiamento parcial para o desenvolvimento desse trabalho.

Enfim, agradeço a todos que conviveram comigo durante esses últimos anos. Aos amigos os quais me esqueci de citar, desculpem-me, além da idade e da falta de cabelo, minha memória não é mais a mesma.

# Lista de Abreviações

AFS - *Avaliador de Fluxos de Saúde*

APDU - *Application Protocol Data Unit*

BAN - *Body Area Network*

BLE - *Bluetooth Low-Energy*

CoAP - *Constrained Application Protocol*

CDG - *Continua Design Guidelines*

CMT - *Controlador do Meio de Transporte*

DPS - *Dispositivo Pessoal de Saúde*

ECG - *Eletrocardiograma*

FHIR - *Fast Healthcare Interoperability Resources*

GATT - *Generic Attribute Protocol*

HCI - *Host Controller Interface*

HL7 - *Health Level 7*

HTTP - *Hyper Text Transfer Protocol*

IEEE - *Institute of Electrical and Electronics Engineers*

IHE - *Integrating the Healthcare Enterprise*

IoT - *Internet of Things*

IoMT - *Internet of Medical Things*

IP - *Internet Protocol*

IPS - *Informação Pessoal de Saúde*

IPSP - *Internet Protocol Support Profile*

IRM - *Interpretador de Regras de Monitoramento*

ISO - *International Standard Organization*

L2CAP - *Logical link control and adaptation protocol*

M2M - *Machine-to-Machine*

MRP - *Monitoramento Remoto de Pacientes*

MQS - *Monitor de QoS para fluxos de Saúde*

MQTT - *Message Queue Telemetry Transport*

ORU - *Observation Response Unit*

PAN - *Personal Area Network*

PRMP - *Processo de Monitoramento Remoto de Paciente*

QoS - *Quality of Service*

RD - *Registro de Dispositivos*

REST - *Representational State Transfer*

SMRP - *Sistema de Monitoramento Remoto de Pacientes*

SMS - *Simple Message Service*

SOAP - *Simple Object Access Protocol*

TCP - *Transport Control Protocol*

TDMA - *Time-Division Multiple Access*

UPnP - *Universal Plug and Play*

URL - *Uniform Resource Locator*

WAN - *Wide Area Network*

# Lista de Figuras

# Lista de Tabelas

# Sumário

# Capítulo 1

# Introdução

Com os últimos avanços tecnológicos em comunicação sem fio e sistemas embarcados, o número de dispositivos conectados em rede é cada vez maior, além disso, estes dispositivos são cada vez menores, autônomos e inteligentes. Portanto, esses dispositivos embarcados, sejam eles sensores ou atuadores, tornam-se capazes de se conectar entre si e com a Internet, viabilizando a chamada Internet das Coisas (do ingês *Internet of Things – IoT*) [1]. A IoT vislumbra uma rede constituída por dispositivos e não apenas por pessoas. Uma tendência natural desses dispositivos na IoT é a sua interação com o mundo físico. Por exemplo, esses dispositivos podem ser sensores que coletam informações do mundo físico, como um sensor de temperatura, mas também podem ser atuadores que alteram o ambiente, como uma válvula que controla a circulação de ar em uma casa. Esse cenário onde sensores, atuadores, e unidades computacionais interagem com o mundo físico são viáveis através de sistemas físico-cibernéticos [2] [3]. Deste modo, com a concretização da IoT, uma tendência é a integração dos sistemas físico-cibernéticos com a Internet.

Dado o cenário apresentado, um dos desafios da IoT com os sistemas físico-cibernéticos está em como fazer dispositivos heterogêneos comunicarem-se de maneira eficaz. Esses dispositivos, de maneira geral, apresentam restrições no que diz respeito ao consumo de bateria, processamento e armazenamento. Com isso, o uso de tecnologias de comunicação eficientes e de baixo consumo é essencial. Seguindo essa linha de avaliação, recentemente têm sido desenvolvidas e aprimoradas tecnologias de comunicação e sensoriamento de baixo custo, manutenção e consumo de energia, como o Bluetooth [1], Near-Field Communication

[1] http://www.bluetooth.org

(NFC) [2], ZigBee [3] e o recente Bluetooth Low Energy (BLE), apenas para mencionar algumas.

Concomitantemente, o crescente número de dispositivos conectados, como *smartphones* e *smartwatches*, viabiliza o desenvolvimento de novos serviços e aplicações, permitindo que a IoT se integre com tecnologias de uso pessoal. Nessa linha de pesquisa e desenvolvimento, novos protocolos estão sendo desenvolvidos e avaliados em diversas camadas. Protocolos como o *Constrained Application Protocol* (CoAP) [4] e o *Message Queue Telemetry Transport* (MQTT) [5] apresentam soluções na camada de aplicação que podem ser utilizadas por dispositivos embarcados com poucos recursos computacionais e de armazenamento energia, como sensores. Em comum, esses protocolos são executados sobre o Protocolo de Internet (do inglês, *Internet Protocol* - IP), viabilizando seu transporte na Internet e com dispositivos pessoais.

Considerando o uso pessoal dessas tecnologias em dispositivos para o consumidor final, a utilização do BLE para comunicação com a Internet torna-se bastante promissora dada sua ampla adoção em dispositivos como *smartphones*, *smart-tvs*, entre outros, além do seu baixo consumo de energia [6]. Alinhado com esse raciocínio, novos perfis ou configurações permitem que novos sensores e dispositivos possam transmitir dados IP utilizando suas interfaces de transmissão. É o caso do perfil *IP Service Profile* (IPSP) do BLE [7] em conjunto com novos mecanismos de controle de fluxo na camada de enlace, os quais criam novos tipos de canais para comunicação para transportar dados IP. Além desse novo tipo de canal, uma especificação para o uso do protocolo 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*) [8] sobre BLE foi definida pelo IETF.

Esse conjunto de tecnologias e dispositivos permite que dispositivos de uso pessoal, como *smartphones*, tornem-se Gateways de Internet para dispositivos periféricos de um usuário utilizando o BLE. Esse cenário permite que Gateways possam ser considerados serviços pessoais, ou seja, são utilizados por um único usuário através de seu dispositivo pessoal. Nesse contexto, esses Gateways criam redes do tipo PAN BLE (do inglês *Personal Area Network* - PAN) e se conectam com diversos dispositivos e serviços. Com essa configuração, uma mesma PAN pode ser compartilhada por serviços de áudio, vídeo, acessibilidade, e interação humano-máquina, como ilustrado na Figura 1.1.

[2] http://www.nfc-forum.org

[3] http://www.zigbee.org

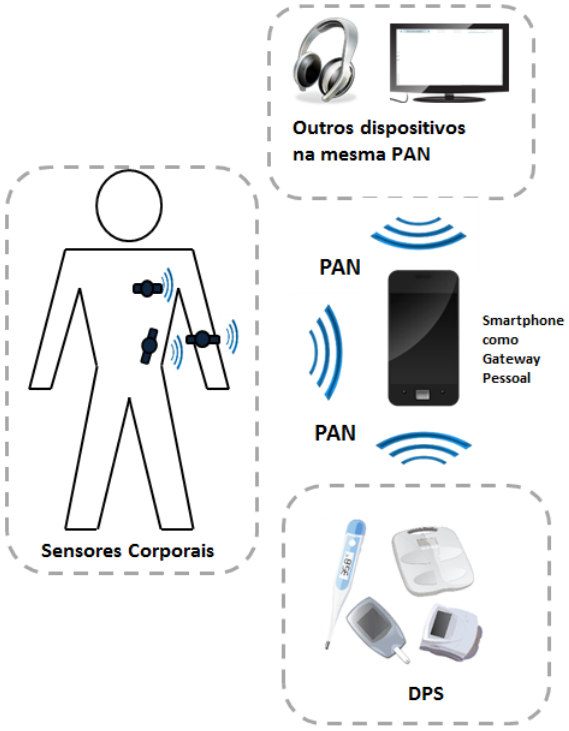


Figura 1.1: Esquema para o compartilhamento da rede PAN e BAN entre diversos dispositivos.

Esse tipo de compartilhamento de rede pode levar a um decréscimo na Qualidade de Serviço (QoS) de rede dos dispositivos periféricos conectados, como uma redução na taxa de transmissão média por exemplo. Considerando redes com tecnologia Bluetooth Low-Energy, por exemplo, o dispositivo mestre da rede é o responsável pelo gerenciamento do fluxo de dados entre os dispositivos clientes. Nesse tipo de rede, portanto, fica a critério do mestre distribuir recursos na rede, ou seja, esse dispositivo acaba distribuindo os recursos de maneira igualitária entre todos os clientes.

Nesse sentido, requisitos de QoS de rede para BLE podem ser definidos a partir de alguns parâmetros, os quais são utilizados pela camada de enlace da especificação Bluetooth [9] para o controle de canais:

- **Taxa de transmissão máxima**: define o quão rápido pacotes devem ser enviados bi-direccionalmente por aplicações BLE.
- **Latência de acesso**: define o atraso máximo aceitável para um pacote ser enviado a

interface aérea após sair da camada de enlace.

- **Variação de atraso**: define os valores máximos e mínimos de atraso de envio de pacotes entre as camadas de enlace de dois nós.

Analisando esses requisitos de QoS em relação a sua percepção para aplicações BLE, a taxa máxima de transmissão torna-se o principal fator de avaliação. Redes PAN BLE, portanto, devem garantir valores de taxa de transmissão para aplicações com diferentes requisitos de QoS. Essas aplicações podem pertencer a diferentes domínios, tais como o domínio de saúde em sistemas de Monitoramento Remoto de Pacientes.

## 1.1 Sistemas de Monitoramento Remoto de Pacientes

Em paralelo ao cenário da IoT, recentemente, com o aumento de custos com a saúde pessoal e a demanda crescente por novos serviços para o tratamento de doenças crônicas, novos desafios e oportunidades também são criados para os serviços de saúde [10]. Neste cenário, o uso da tecnologia para o monitoramento de pacientes vem crescendo ano após ano. Esse interesse por tecnologias de monitoramento de saúde impulsiona o desenvolvimento de novos Dispositivos Pessoais de Saúde (DPS) com interfaces de comunicação embutidas. Exemplos de DPS incluem medidores de pressão arterial ou glicosímetros.

Com esses DPS, dados pessoais de saúde são coletados e enviados para a Internet através de suas interfaces de comunicação. Deste modo, profissionais da área de saúde podem monitorar a evolução do estado de saúde dos pacientes remotamente, e tomar ações com antecedência, evitando complicações futuras. Este processo, em conjunto com suas tecnologias envolvidas, é conhecido como Saúde Conectada [11]. Em termos gerais, um sistema de Saúde Conectada viabiliza um cenário onde DPS se conectam com a Internet para exportar seus dados, portanto, construindo a Internet das Coisas para a área de saúde. Podem-se vislumbrar casos de uso onde um DPS faz a coleta dos dados de saúde do paciente, e estes dados são compartilhados automaticamente com o seu médico transparentemente.

Um sistema de Saúde Conectada é composto por múltiplos componentes, desde o DPS até o serviço de Monitoramento Remoto de Pacientes (MRP) na Internet. Em termos gerais, o primeiro passo nesse sistema é a coleta de Informações Pessoais de Saúde (IPS)

através de um DPS, e o seu compartilhamento pela Internet através de uma interface de comunicação. Por exemplo, a realização de uma coleta de dados de pressão arterial através de um tensiômetro com tecnologia Bluetooth. A partir dessa coleta e compartilhamento de dados, um dispositivo agregador, ou um Gateway, recebe essas IPS e as encaminha para o serviço de MRP na Internet.

Como introduzido anteriormente, esses Gateways podem ser dispositivos pessoais portáteis, como *smartphones* ou computadores pessoais. Portanto, desde o DPS até o serviço em nuvem, a informação é transportada por diversos meios de comunicação até o seu destino, desde uma rede pessoal ou corporal (do inglês *Body Area Network* - BAN) até uma rede de larga escala (do inglês *Wide Area Network* - WAN) como a Internet.

Alguns desafios devem ser considerados na implantação de sistemas de Saúde Conectada para o MRP. A depender do público alvo, diferentes tipos de dispositivos poderão ser utilizados. Por exemplo, se o caso de uso for a realização de um MRP contínuo em diferentes localizações, o uso de Gateways portáteis e pervasivos, como *smartphones*, torna-se obrigatório. Em conjunto com esses Gateways, sensores corporais podem ser utilizados como DPS, como por exemplo, uma pulseira que faz aferições de frequência cardíaca continuamente. Do ponto de vista tecnológico, além de utilizar diferentes tecnologias de transmissão, como BLE, esses dispositivos e Gateways precisam definir protocolos para a troca de dados de saúde. Em relação a esse ponto, boa parte das soluções e fabricantes definem seus próprios protocolos, criando soluções verticais onde seus DPS conversam apenas com seus Gateways e serviços de saúde.

Em relação a esses desafios, associações e grupos de trabalhos definiram padrões de interoperabilidade para diversos níveis da cadeia de comunicação de um sistema de Saúde Conectada. No nível de DPS, na família de padrões ISO/IEEE 11073 define-se como esses dispositivos devem trocar dados com outras entidades (o padrão ISO/IEEE 11073:20601 [12]), e como as informações de saúde devem ser representadas (o padrão ISO/IEEE 11073:10101 [13]). Um detalhe do ISO/IEEE 11073 é sua independência da camada de transporte, viabilizando seu uso sobre qualquer tipo de protocolo ou tecnologia de transporte. Em relação ao compartilhamento de dados de saúde com serviços na Internet, o Continua Health Alliance[4], apresenta recomendações para o compartilhamento de dados de saúde entre DPS, agregado-

[4]http://www.continuaalliance.org

res de dados e serviços de armazenamento de dados de saúde [11]. Essas recomendações têm como objetivo viabilizar um cenário onde DPS compartilham dados com a Internet através de Gateways e agregadores padronizados.

Em um cenário mais amplo da Saúde Conectada, pacientes são monitorados constantemente, e processos de MRP (PMRP) podem ser executados a qualquer momento. Um PRMP se caracteriza como uma sequência de ações onde dispositivos e sensores enviam dados a serviços na Internet, os quais tomam decisões sobre a saúde do paciente.

Voltando ao cenário de MRP, na maioria dos casos esse decréscimo nas características de QoS entregues não influencia nas aplicações. Por exemplo, uma aferição eventual de pressão arterial pode ser entregue com alguns segundos de atraso. Entretanto, durante situações de urgência onde as informações de um DPS são essenciais para um diagnóstico, torna-se necessário elevar os requisitos de QoS desse DPS durante o processo de monitoramento. Por exemplo, em um processo de MRP, diversos dispositivos podem ser utilizados, como um monitor de ECG portátil, um medidor contínuo de glicose implantado sob a pele, e bombas de insulina, e esses dispositivos devem ter seus requisitos de QoS de rede satisfeitos em prol do bem-estar do paciente.

É nesse contexto, de compartilhamento de recursos em uma rede PAN entre dispositivos na IoT no qual se insere esse trabalho. Mais especificamente, propõe-se uma abordagem para o controle do fluxo de dados em Gateways pessoais, onde a informações enviadas por aplicações são utilizadas para a definição dos dispositivos participantes em um processo de monitoramento. Essas informações caracterizam um *Contexto de Uso*, o qual é a representação de uma situação onde um conjunto de dispositivos trabalha junto realizando uma sequência de ações em prol de um processo mais amplo.

Mais especificamente, neste trabalho é apresentada uma abordagem para o controle de fluxo de dados em redes pessoais do tipo BLE para aplicações na IoMT (do inglês, *Internet of Medical Things*), as quais fazem parte de Sistemas de Monitoramento Remoto de Pacientes. Esse controle é realizado de maneira adaptativa, onde os recursos de rede são distribuídos dinamicamente através do tempo a depender da prioridade e característica de cada dispositivo. Essa distribuição adaptativa realiza um controle da taxa de transmissão entre os dispositivos da rede PAN, desse modo fornecendo os requisitos de QoS aos dispositivos desejados.

## 1.2 Problemática e Justificativa

Redes do tipo PAN ou BAN que utilizam Gateways em dispositivos pessoais, como *smartphones* ou *tablets*, podem ter aplicações para diversos fins, como entretenimento, através de periféricos de áudio e vídeo, acessibilidade através de *smart-glasses*, e cuidados com a saúde através de sensores corporais e DPS. Com o advento de novos perfis e protocolos para Internet voltados para tecnologias de baixo consumo, como o IPSP [7] e a adaptação do 6LoWPAN para o Bluetooth Low Energy (BLE) [8], novos serviços vão ser adicionados a essas redes, deixando-as cada vez mais congestionadas. Redes de baixo consumo, como o BLE, apresentam diversas vantagens em relação ao consumo de energia, simplicidade e disponibilidade em dispositivos pessoais e domésticos [14]. Entretanto, dada sua simplicidade, conexões através de sua camada de enlace são tratadas como sendo de *best-effort*, como descrito na especificação do Bluetooth 4.2 [9]. Ou seja, todas as conexões são tratadas de maneira igual, não sendo priorizado nenhum dispositivo em relação ao outro em termos dos requisitos de QoS apresentados anteriormente.

Essa limitação na distribuição de recursos em redes Bluetooth Low-Energy com suporte a IPv6 existe devido ao novo modelo de controle de fluxo baseado em créditos introduzido na especificação Bluetooth 4.2. Nessa especificação não são definidos modos de priorização entre dispositivos clientes para garantia de QoS. Portanto, implementações da especificação BLE com controle de fluxo baseado em créditos fazem uso de um modelo simples de distribuição de recursos (créditos), criando assim canais *best-effort*.

Outra limitação em redes BLE vem do hardware ou firmware Bluetooth. A depender do hardware do controlador Bluetooth, limitações relativas ao número de dispositivos conectados e a taxa de dados máxima suportada pelo dispositivo mestre surgem. Comumente, controladores Bluetooth não conseguem lidar com uma taxa de dados maior do que o seu projeto suporta e, ao invés de rejeitar novas conexões, o controlador para de funcionar e entra em um estado instável. Portanto, em situações como essa, o dispositivo mestre deve realizar um controle de fluxo adequado de modo a não ultrapassar seus limites e evitar desconexões na rede PAN.

No âmbito de aplicação desse trabalho, Sistemas de Monitoramento Remoto de Pacientes (SMRP) são avaliados para a Internet das Coisas. Em especial, é avaliado o compartilha-

mento da rede PAN BLE entre DPS e outros dispositivos e serviços. Considera-se que DPS participantes de um SMRP não são utilizados constantemente em processos de interesse do usuário, ou Processos de Monitoramento Remoto de Pacientes (PMRP). Por exemplo, em algumas situações o paciente ou um serviço de monitoramento na Internet tem apenas interesse em ter acesso à última aferição de pressão arterial coletada pelo DPS, e esse tipo de evento não necessita propriamente de requisitos de QoS especiais.

Entretanto, considerando o exemplo anterior, a depender do valor de pressão arterial coletado, o serviço de monitoramento na Internet poderá requisitar mais informações para uma melhor avaliação do paciente e, consequentemente, uma melhor tomada de decisão para situações de emergência. Por exemplo, ao receber um valor de pressão arterial considerado alto para o perfil do paciente, o serviço monitoramento na Internet pode requisitar um fluxo de dados de ECG e de dados relativos aos sensores de movimento do paciente. Essa nova sequência de dados irá ser enviada através do Gateway pessoal, o qual pode estar sendo compartilhado por outros dispositivos e serviços. Como descrito anteriormente, todas as conexões são tratadas como de *best-effort*, e a depender da capacidade da rede, os DPS e sensores participantes do PMRP podem ser afetados. Por exemplo, o DPS de ECG pode não conseguir transmitir os dados da curva de ECG com a taxa de transmissão necessária ao serviço monitoramento na Internet devido a um gargalo no primeiro nível de rede, a rede PAN.

Portanto, além de controlar o fluxo de dados entre todos os dispositivos da rede PAN, é importante identificar o contexto de uso ou situação pela qual o sistema ou usuário está passando para identificar e tratar as conexões dos dispositivos em relação aos seus requisitos de QoS e suas prioridades. Ou seja, considerando o exemplo de sistema de Monitoramento Remoto de Pacientes, ao identificar que um paciente está ou irá passar por um PMRP, o Gateway deve garantir os requisitos de QoS dos dispositivos participantes durante aquele período de monitoramento.

No contexto desse trabalho, portanto, um Gateway capaz de identificar a situação pela qual o paciente está passando, e que seja capaz de realizar as adaptações necessárias no fluxo de dados, é considerado um *Smart-Gateway*. Para o *Smart-Gateway* alguns requisitos devem ser considerados, de modo que o mesmo seja capaz de:

- avaliar o tráfego de rede de maneira autônoma com o objetivo de identificar o contexto

de uso, monitoramento ou a situação atual do sistema ou usuário;

- identificar quais serão os dispositivos participantes de um processo de monitoramento, dadas informações de camadas superiores ou dada a situação atual do usuário ou do dispositivo;

- adaptar os fluxos de dados para os dispositivos participantes de um processo de monitoramento baseado em seus requisitos e no estado atual da rede.

Alguns trabalhos exploraram características de redes Bluetooth Low-Energy, mostrando suas características de pouco consumo de energia e limitações na taxa de transmissão. Trabalhos como os apresentados em [14], Nieminen et al. [6] e Siekkinen et al. [15] apresentaram avaliações de desempenho de redes Bluetooth Low-Energy em diferentes situações. Entretanto, dentre os trabalhos pesquisados, não foi encontrado um que realize uma aplicação de controle de fluxo adaptativo para redes BLE. Além disso, nenhum trabalho explorou o novo modelo de controle de fluxo baseado em créditos apresentado no Bluetooth 4.2.

Em relação à Sistemas de Monitoramento Remoto de Pacientes, diversos trabalhos apresentam soluções e sistemas de MRP para o compartilhamento de IPS na Internet. Em [16], [17] e [18] são discutidos aspectos e oportunidades relacionadas à integração entre diferentes domínios, incluindo cuidados com a saúde e a indústria de *Consumer Electronics*. Outros trabalhos apresentam soluções de integração para domínios específicos, como em [19], onde um sistema foi apresentado para a integração de DPS com aparelhos de TV através de uma plataforma de interação MHP (Multimedia Home Platform). Em outro trabalho é apresentada uma solução para a integração de SMRP com redes sociais [20]. Outra linha de pesquisa e desenvolvimento apresenta proposta de *middlewares* para a utilização em sistemas e aplicações para o cuidado com a saúde, como nos trabalhos apresentados em [21], [22] e outros listados em [23]. Outros trabalhos apresentam propostas e arquiteturas de infraestrutura para redes de sensores voltadas para o MRP e sua integração com a Internet, como os trabalhos em [24], [25] e [26]. Em comum, nenhum dos trabalhos apresentados anteriormente apresenta uma arquitetura padronizada para a transmissão de dados de saúde entre DPS e serviços de MRP na Internet, onde seja realizado um controle de fluxo de dados baseado no contexto de uso fornecido pela camada de aplicação.

Alguns outros trabalhos, entretanto, foram desenvolvidos em relação ao controle de fluxo de dados de saúde em redes sem fio em um Sistema de Monitoramento Remoto de Pacientes. Em [27] e [28] são apresentados mecanismos para controle de fluxo para dados de saúde entre um Gateway de Saúde e o serviço de MRP na Internet. Em [29] é apresentado um mecanismo de escalonamento de tráfego com o intuito de priorizar tráfegos de monitoramento de saúde também entre o Gateway de Saúde e o Serviço de MRP na Internet. Nesses trabalhos não são abordados meios para o controle de fluxo de dados na rede PAN ou BAN. Outros trabalhos focaram na camada de Controle de Acesso ao Meio (MAC) em redes BAN. Em [30] foi apresentado um sistema que faz uso de uma camada MAC virtual para ajustes em parâmetros de QoS para sensores na BAN através de um mecanismo de *polling*. Em [31] também é apresentado um mecanismo independente de meio de transporte baseado em *polling* para redes BAN. Apesar de atuarem na rede BAN (ou PAN), esses trabalhos fazem uso de mecanismos de *polling*, ou seja, o Gateway que é responsável pelas requisições de dados para os sensores. Esse tipo de mecanismo faz com que os DPS ou sensores sejam dependentes do serviço de monitoramento. Em várias situações, entretanto, o sensor deve ter autonomia para iniciar a transmissão quando perceber que dados estão disponíveis. Para essas situações, o trabalho [32] apresenta uma proposta onde o controle de fluxo utiliza modelo baseado em tempo (TDMA) onde novos slots são alocados aos sensores em estado de emergência. Entretanto, nenhum desses trabalhos apresenta uma abordagem padronizada para aferição de contexto para a aplicação no controle de fluxo de dados de saúde através da avaliação de tráfego. Em todos os trabalhos consideram-se apenas que determinados fluxos têm mais prioridade do que outros.

Portanto, apesar da grande quantidade de trabalhos na área de Monitoramento Remoto de Pacientes e controle de fluxo de dados para dispositivo em redes pessoais, nenhum dos trabalhos apresenta uma solução completa onde a avaliação de contexto em nível de aplicação é aplicada para a adaptação de fluxos de monitoramento no Gateway. Ou seja, uma solução que permita que Gateways possam realizar um controle de fluxo de dados adequado em sua rede PAN para diferentes contextos de aplicação. Por fim, como introduzido anteriormente, também não foram encontrados trabalhos que explorassem o novo mecanismo de controle de fluxo do Bluetooth Low-Energy para redes IPv6.

## 1.3 Objetivos

O principal objetivo dessa tese é apresentar uma abordagem para o controle de fluxo de dados em Gateways BLE utilizando informações obtidas a partir de aplicações, as quais podem identificar contextos de uso através da análise do tráfego de dados no próprio Gateway. Em especial, o fluxo de dados no contexto da IoMT foi avaliado. Dentre os objetivos específicos alcançados destacam-se:

1. Definir e implantar uma infraestrutura de rede padronizada para possibilitar o Monitoramento Remoto de Pacientes na Internet das Coisas, ou a IoMT.

2. Definir uma abordagem para a análise do tráfego de Dispositivos Pessoais de Saúde com o propósito de fornecer dados para a avaliação de situações de maneira transparente.

3. Definir e avaliar um controlador adaptativo do fluxo de dados em Gateways Bluetooth Low-Energy utilizando informações de contexto e de aplicações externas.

4. Implantar o controle de fluxo adaptativo em uma rede PAN do tipo Bluetooth Low-Energy para Sistemas de Monitoramento Remoto de Pacientes.

5. Realizar uma avaliação experimental do controlador proposto em relação a situações de monitoramento, de modo a validar que a abordagem proposta viabiliza o uso de DPS em redes PAN de maneira eficaz.

Considerando os objetivos específicos, algumas características e definições precisam ser consideradas para o desenvolvimento do controle de fluxo de dados baseado em contexto. Para a definição da arquitetura do sistema de Monitoramento Remoto de Pacientes foi utilizado o padrão ISO/IEEE 11073 como base para o modelo de dados de saúde trafegados [12]. No contexto deste padrão são definidas características que permitem sua transmissão utilizando diferentes protocolos, e são definidos mecanismos para a descrição de vários dados referentes a sinais vitais (saúde). A utilização do ISO/IEEE 11073 como base também possibilita a avaliação dos dados de contexto de maneira padronizada. Portanto, o Gateway pessoal tem uma linguagem padrão para a análise do tráfego para aferição de situações de monitoramento.

Para o transporte dos dados ISO/IEEE 11073 utiliza-se o protocolo CoAP como base [4]. O CoAP oferece mecanismos para troca de dados na Internet de maneira semelhante a outros protocolos, como o HTTP. Portanto, sua utilização permite que os dados de saúde trafeguem em diferentes redes através da utilização de proxies [33], e esses proxies podem ser utilizados para análise de dados. A utilização e avaliação do CoAP em conjunto com ISO/IEEE 11073 foi realizado em trabalhos anteriores desenvolvidos durante esse trabalho de tese [34], comprovando suas características para dispositivos limitados em comparação com outros protocolos, como o TCP/IP.

Para o desenvolvimento e definição do controle de fluxo de dados baseado em contexto foi utilizada a camada de enlace do Bluetooth Low-Energy (L2CAP) como ponto de controle [9]. Essa camada oferece o mecanismo base para o controle do fluxo de dados baseado em créditos. Com isso, para a definição do modelo de controle adaptativo foi alterado o modelo de controle de fluxo de dados do BLE em uma plataforma Linux, a partir da implantação de um novo modelo, também baseado em créditos, o qual foi aplicado diretamente ao módulo de controle Bluetooth do Linux, o *BlueZ*[5].

## 1.4 Contribuições

Durante o desenvolvimento desse trabalho de tese, atividades adicionais foram desenvolvidas na mesma linha de pesquisa. Um trabalho relacionado à integração de sistemas de MRP com dispositivos pessoais em redes locais através de uma infraestrutura *Universal Plug and Play* (UPnP) foi proposto e avaliado [35]. A partir dos resultados e da experiência adquirida com esse trabalho, contribuições foram feitas para a especificação *IoT Management and Control* [36] do UPnP Forum[6]. Essa especificação é utilizada como base da iniciativa UPnP para Internet das Coisas[7]. Em outra linha de pesquisa e desenvolvimento relacionada ao trabalho de tese, o conhecimento adquirido em agregadores de saúde, seus modelos de comunicação, e seus modos de uso, contribuíram para a criação e aplicação de uma patente no *United States Patent and Trademark Office* (USPTO) [37]. Por fim, um capítulo de livro com detalhes arquiteturais de um sistema de saúde conectada padronizado foi escrito e publicado em

[5]http://www.bluez.org

[6]http://www.upnp.org

[7]http://upnp.org/latestupdates/IOTCloud/

conjunto com outros autores [38].

Mais especificamente no contexto dessa tese, outros trabalhos foram realizados para a avaliação do protocolo ISO/IEEE 11073 no contexto da Internet das Coisas [39], onde uma classificação de DPS foi apresentada e discutida. Também foram realizados trabalhos de avaliação e integração do protocolo CoAP com o ISO/IEEE 11073 [34], e sua implantação em um ambiente em nuvem [40]. O conjunto dos resultados obtidos foi utilizado na definição e implantação de uma infraestrutura de Saúde Conectada para a Internet das Coisas, a qual é descrita em um trabalho publicado em periódico [41]. Além disso, até a publicação dessa tese, um novo artigo relativo ao *Smart-Gateway* para Saúde foi aceito para publicação.

Em relação ao controle de fluxo adaptativo aplicado a Gateways Bluetooth Low-Energy, dentro da literatura avaliada, não foram encontrados trabalhos que explorassem o novo mecanismo de controle baseado em créditos do Bluetooth Low-Energy para dados IPv6. Portanto, o projeto e avaliação de modelos de controle aplicados a esse novo mecanismo baseado em créditos torna-se uma contribuição inédita até a publicação dessa tese. A partir dessa nova proposta de controle adaptativo no Bluetooth Low-Energy, novas pesquisas estão sendo desenvolvidas no Laboratório de Sistemas Embarcados e Computação Pervasiva da Universidade Federal de Campina Grande, as quais abrangem a avaliação dos modelos de controle através de métodos formais, e a criação de novas aplicações para avaliação de contexto em diferentes domínios além do de saúde.

## 1.5 Organização do Documento

Esse documento está organizado da seguinte maneira:

- no Capítulo 2 são apresentadas as tecnologias e conceitos relacionados ao trabalho desenvolvido, de modo a facilitar o entendimento do leitor em relação as escolhas realizadas durante o seu desenvolvimento;
- no Capítulo 3 são apresentados trabalhos relacionados as diversas áreas abordadas nesse trabalho;
- no Capítulo 4 é apresentado detalhes da arquitetura base do Sistema de Monitoramento Remoto de Pacientes para a Internet das Coisas desenvolvido neste trabalho. Esse sis-

tema é a base para o desenvolvimento do objetivo principal desse trabalho. Resultados experimentais são apresentados de modo a validar a arquitetura proposta e a relevância do problema proposto nesse trabalho;

- no Capítulo 5 é detalhado o projeto do controlador adaptativo de fluxo de dados aplicado a Gateways BLE. Neste capítulo foram apresentados detalhes sobre as limitações do BLE e como o controlador adaptativo pode ser utilizado para contornar esses problemas. Diferentes modelos de controles foram apresentados durante o projeto do controlador.

- no Capítulo 6 são apresentados detalhes sobre os experimentos de validação do controlador adaptativo BLE em diferentes situações.

- no Capítulo 7 é detalhado o processo de implantação e avaliação de um *Smart-Gateway* para Saúde. O principal objetivo do *Smart-Gateway* é realizar o controle de fluxo de dados de saúde baseado em informações de aplicações e avaliação do tráfego de dados.

- no Capítulo 8 são apresentadas as conclusões sobre os resultados obtidos e as perspectivas de trabalhos futuros.

# Capítulo 2

# Fundamentação Teórica

Neste capítulo são apresentados detalhes sobre a tecnologia Bluetooth Low-Energy e suas características que permitem uso de conexões IP sobre seus canais de comunicação. Em especial é detalhado o funcionamento do canal com controle de fluxo baseado em créditos para o transporte de pacotes IP utilizando o BLE. Também são apresentados nesse capítulo conceitos relacionados aos sistemas de Monitoramento Remoto de Pacientes (MRP) e a Internet das Coisas, os quais são considerados importantes para o entendimento da solução apresentada nesse trabalho. Mais especificamente, são apresentados detalhes dos principais protocolos utilizados durante a definição e desenvolvimento da solução.

## 2.1 Bluetooth Low-Energy e o 6LoWPAN

Semelhantemente ao Bluetooth convencional (BR/EDR), o BLE opera na faixa de frequência ISM de 2.4GHz [9]. O BLE também faz uso de *frequency hopping*, e oferece dois esquemas para múltiplo acesso à camada física, o *Frequency Division Multiple Access* (FDMA) e *Time Division Multiple Access* (TDMA). Quarenta (40) canais são utilizados no esquema FDMA, dos quais, três (3) são utilizados para *advertising* e 37 para troca de dados. O esquema TDMA faz com que cada dispositivo tenha um tempo pré-determinado para a transmissão de pacotes. O canal físico, então, é dividido em unidades de tempo conhecido como eventos de conexão (*connection events*). Dados são transmitidos entre os dispositivos BLE dentro desses eventos. Após a conexão ser estabelecida, o iniciador se torna o mestre (*master*) de uma rede *piconet*, e o outro dispositivo se torna seu cliente (*slave*), como ilustrado na Figura

2.1.

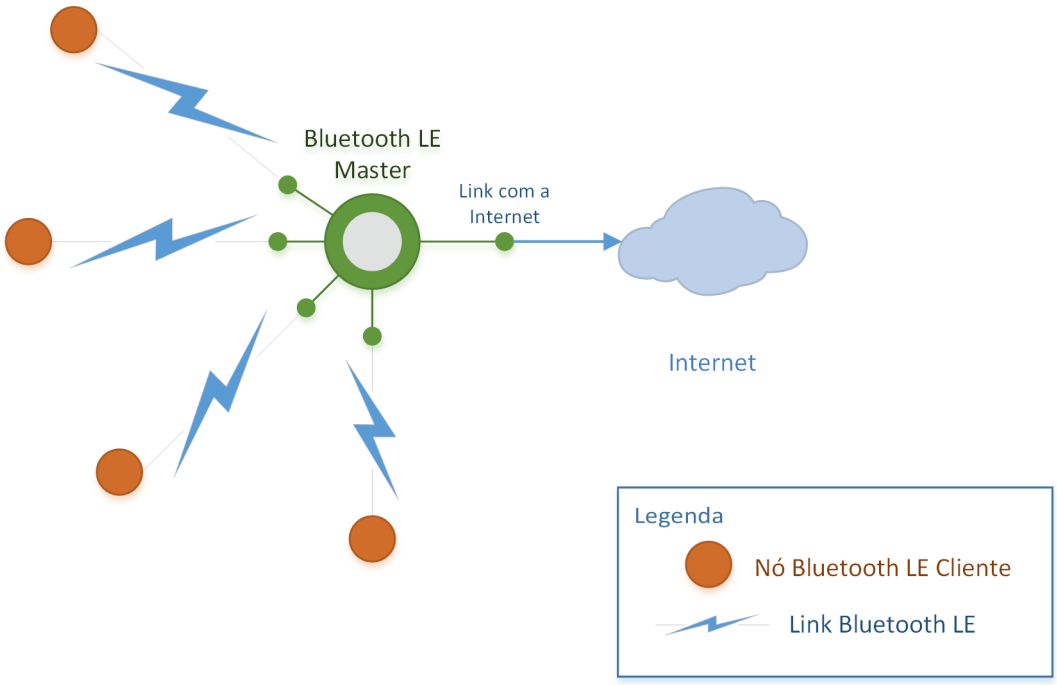


Figura 2.1: Formação de uma rede Piconet BLE.

Sobre o canal físico existem os conceitos de enlace, canais e protocolos de controle. A hierarquia é composta pelo canal físico, enlace físico, transporte lógico, enlace lógico e canal L2CAP (*Logical Link Control and Adaptation Protocol*). Por exemplo, dentro de um canal físico, um enlace físico é formado entre o mestre e o cliente. Um protocolo de controle para as camadas de enlace e física é executado dentro de enlaces lógicos. Esse é o protocolo da camada de enlace (LL). A camada de enlace utiliza o protocolo LL para controlar os dispositivos na *piconet* e oferecer serviços de gerenciamento para camadas mais baixas, como a camada física. Eventos de conexão são utilizados para a troca dados entre o mestre e os clientes. Todo evento de conexão inicia quando um pacote é transmitido a partir do mestre. Quando o cliente recebe o pacote, o mesmo deve enviar uma um pacote de volta ao mestre. Esse pacote pode ser de dados ou apenas um reconhecimento de recebimento (*ack*). Ao mestre, entretanto, não é requerido enviar de volta outro pacote, ficando ao seu critério continuar ou não o evento de conexão.

A cada evento de conexão, mestre e cliente usam um novo canal de frequência, seguindo as regras do algoritmo de mudança de frequência (*frequency hopping*). O tempo entre o início de dois eventos de conexão consecutivos é definido pelo parâmetro *connInterval*, o

qual é um múltiplo de 1.25ms entre o intervalo de 7.5ms e 4s. Outro importante parâmetro, o *connSlaveLatency*, define o número de eventos de conexão consecutivos ao qual o cliente não necessita esperar por pacotes do mestre, e portanto, pode deixar seu rádio desligado. Além desses parâmetros, é definido o *connSupervisionTimeout*, o qual define um período máximo que o cliente deve esperar por eventos de conexão do mestre. Essa definição de parâmetros de tempo, principalmente o *connInterval*, apresenta um mecanismo de controle TDMA que a camada de LL faz uso.

É interessante observar que tanto a camada física quanto a camada de enlace fazem parte do lado controlador de um dispositivo Bluetooth. O lado controlador, fazendo uma definição simples, pode ser considerado a parte de hardware do dispositivo Bluetooth propriamente dito. Por exemplo, um adaptador USB ou PCI Bluetooth é o controlador, enquanto o computador pessoal é o hospedeiro. Nessa divisão, a pilha de protocolos Bluetooth acaba sendo definida por uma interface HCI (*Host Controller Interface*), onde parte do protocolo é controlado pelo hospedeiro, e a outra parte pelo controlador, como ilustrado no diagrama da Figura 2.2.

Nessa divisão de protocolos, é interessante observar que o controle de enlaces (*enlaces* entre dispositivos) é dividido em duas camadas, a camada de enlace no controlador e a camada L2CAP no hospedeiro. Com isso, o hospedeiro pode atuar no controle de canais de maneira independente ao controlador, o qual, normalmente, tem limitações de memória e processamento para realizar esse controle de maneira inteligente. Ou seja, do mesmo modo que o BR/EDR, sobre a camada de enlace, o L2CAP oferece uma abstração de canais para aplicações e serviços. O L2CAP é responsável pela fragmentação e de-fragmentação de dados para aplicações, e a multiplexação e de-multiplexação de canais virtuais dentro de um mesmo enlace lógico. O L2CAP tem um protocolo de controle de canais, onde esses canais podem funcionar dentro de cinco (5) modos. Um desses modos é o de Controle de Fluxo LE baseado em Créditos.

O modo de Controle de Fluxo LE baseado em créditos é utilizado para a criação de canais orientados a conexão para o L2CAP do BLE, onde um esquema de controle baseado em créditos é utilizado no controle de fluxo de dados. Esse é o único modo para a criação de canais orientados a conexão no L2CAP do BLE. Para suportar esse modo, um pacote L2CAP foi definido, o *LE-Frame*, o qual equivale a uma unidade de crédito utilizado no controle

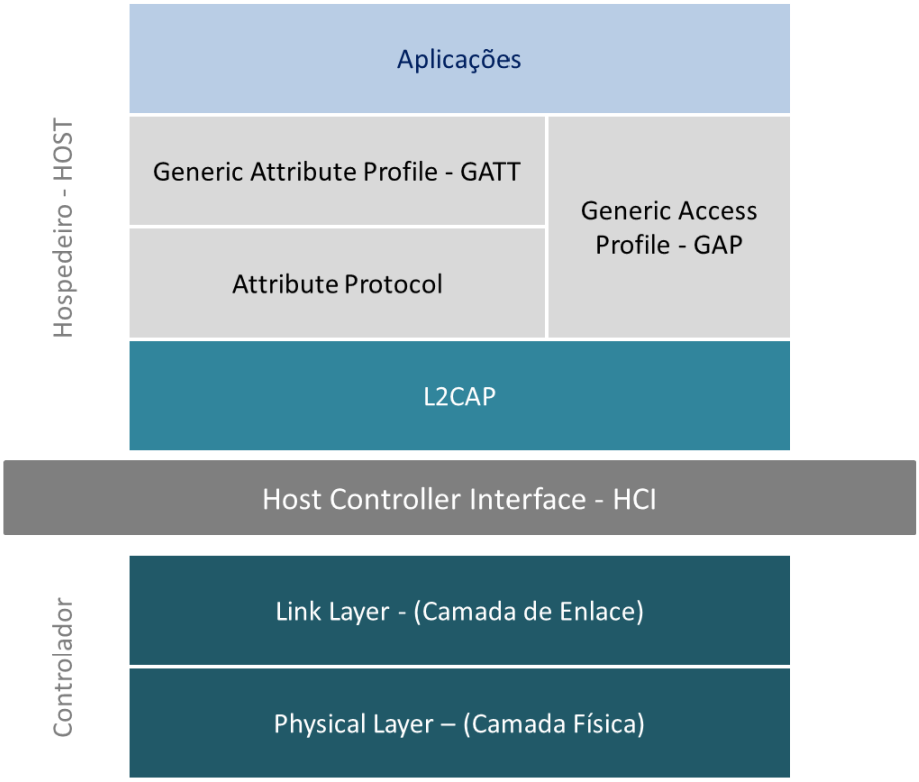


Figura 2.2: Pilha de protocolos para o BLE.

fluxo. Portanto, para realizar o controle de fluxo de dados entre diversos dispositivos, o mestre da *piconet* deve realizar um controle de créditos entre os diferentes clientes da rede como detalhado nas próximas seções.

A partir desses canais orientados a conexão no BLE foi possível realizar uma adaptação do protocolo 6LoWPAN para BLE [8], onde foi especificado como *LE-Frames* transportam pacotes IPv6 sobre esses canais orientados a conexão. Essa especificação, portanto, através de apenas uma atualização na pilha L2CAP permitiu que conexões IPv6 fossem realizadas no BLE sem que os controladores atuais (hardware) forem alterados.

### 2.1.1 Controle de Fluxos Baseado em Créditos na Camada L2CAP

Esse novo modelo de controle de fluxo de dados introduzido no L2CAP permite ao mestre determinar quando e a quantidade de dados que um cliente pode enviar. O conceito de crédito serve como uma permissão para o envio de *LE-Frames* entre os dispositivos. Ao enviar um número $X$ de créditos a outro nó, o nó concedente está permitindo que o nó cliente envie um

número *X* de *LE-Frames* de volta quando esse achar necessário. Para esse funcionamento, tanto o nó concedente quanto o nó que recebe os créditos mantém uma contagem do número total de créditos. A cada *LE-Frame* trocado a contagem é diminuída de um.

A Figura 2.3 apresenta um exemplo de fluxo de troca de dados em um canal orientado a conexão com controle de fluxo baseado em créditos. Nesse exemplo podem-se observar dois tipos de fluxos:

- Fluxo bidirecional entre mestre e cliente, onde as duas partes trocam *LE-Frames* de dados.
- Fluxo de envio de créditos extras, onde uma das partes decide enviar mais créditos a outra parte.

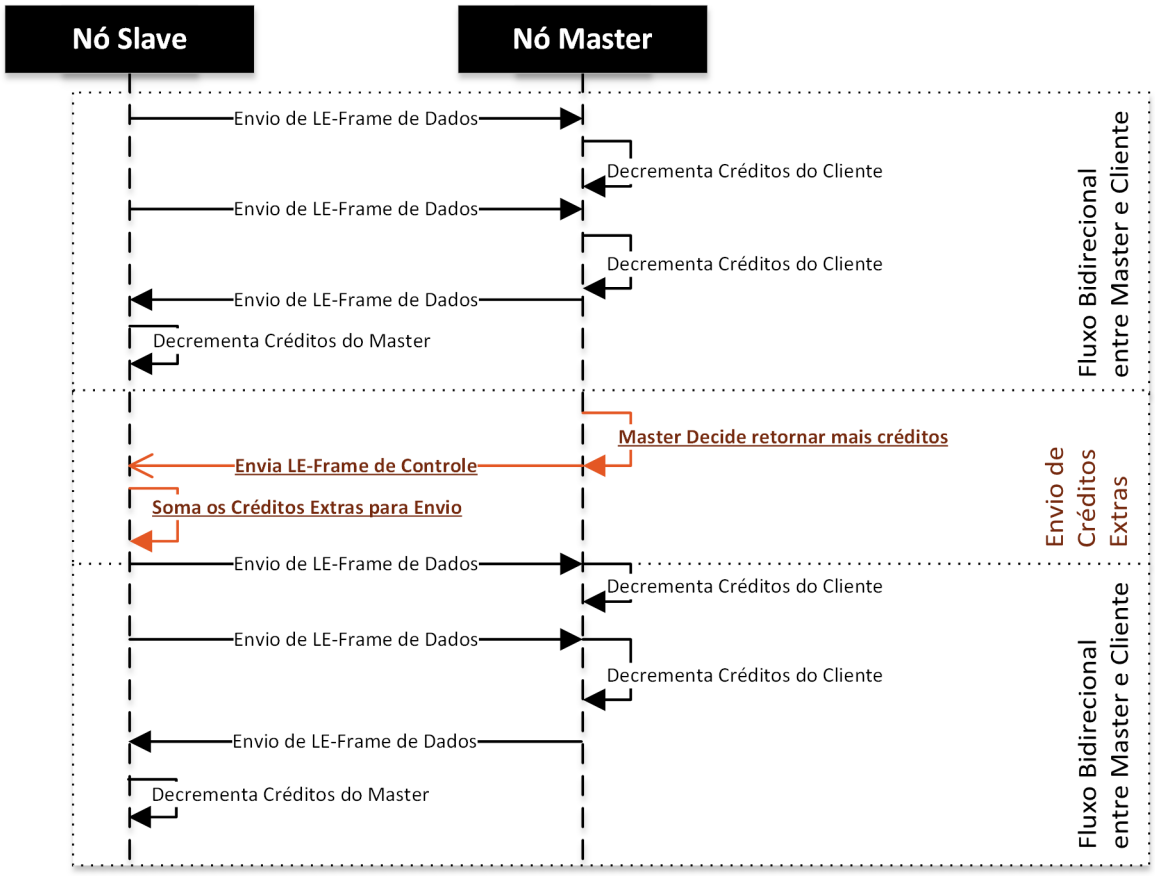


Figura 2.3: Fluxo de dados e controle de créditos na camada L2CAP.

No fluxo bidirecional convencional, toda vez que uma das partes envia um pacote *LE-Frame* para a outra parte é realizado o decréscimo da contagem de créditos de envio, como

apresentado na primeira parte do fluxograma. A critério do nó que recebe dados, nesse caso o mestre, este pode decidir enviar mais créditos de recepção a outra parte. Ao enviar um pacote de controle com mais créditos a outra parte, esse número de adicional de créditos é somado na parte receptora, como apresentado na parte central do fluxograma. Após esse envio de créditos extras, o fluxo de trocas de dados bidirecional é retomado.

Algumas observações devem ser feitas nesse tipo de canal, por exemplo, tanto os pacotes de dados quanto os pacotes de sinalização de créditos utilizam o mesmo meio. Portanto, ao enviar mais créditos para outro nó, o nó concedente deixa de enviar um *LE-Frame* de dados para enviar um pacote de sinalização. Portanto, a troca de créditos entre nós influencia na taxa de dados total em nível aplicação.

É importante observar que esse nível de controle de fluxo na camada L2CAP funciona independentemente do controle de intervalos na camada de enlace. Mesmo quando um nó não tenha créditos para envio de dados, seus intervalos de conexão na camada de enlace serão mantidos e, portanto, uma conexão lógica na camada enlace existe. Esse tipo de controle independente entre as camadas permite que a camada superior no hospedeiro realize processos de controle para sobrepor as limitações da camada inferior no controlador, como será apresentado no Capítulo 5.

Apesar de especificar o uso de canais orientados a conexão com um controle de fluxo baseado em créditos, o L2CAP do BLE não define uma politica para distribuição de créditos. Essa política, caso existente, fica a critério do desenvolvedor do hospedeiro BLE.

### 2.1.2 Perfil GATT e o Protocolo ATT

O *Generic Attribute Profile* (GATT) faz um uso de um protocolo de atributos (ATT) para realizar a troca de informações armazenadas em dispositivos. O ATT define dois papéis, cliente e servidor, e realiza a comunicação entre esses dispositivos através de um canal L2CAP dedicado. O servidor disponibiliza um conjunto de atributos que podem ser acessados por um cliente. Operações de escrita, leitura e notificação são oferecidas pelo ATT. Com o perfil GATT, um dispositivo é capaz de descobrir serviços e realizar a troca de características. Características são organizadas em valor e propriedades. Serviços e características GATT são armazenados em atributos do protocolo ATT. Portanto, o GATT realiza apenas a definição de como identificar um serviço ou característica através do ATT [9].

Perfis GATT são utilizados para a definição de vários serviços e perfis para dispositivos simples. Por exemplo, existem definições BLE GATT para monitores cardíacos, termômetros, balanças, entre outros [1]. Apesar do modo de comunicação IPv6 no BLE fazer uso de um canal L2CAP diferente do utilizado pelo ATT, o primeiro passo na criação de um canal IPv6 no BLE é realizar a descoberta do serviço IP no dispositivo remoto. Essa descoberta é realizada através do GATT, e após isso, o canal dedicado é criado entre os dispositivos, como descrito a seguir.

### 2.1.3 Suporte ao IPv6 e o Perfil de Suporte ao IP – IPSP

Além da criação de canais orientados a conexão no L2CAP do BLE, um novo perfil para suporte ao protocolo IP foi criado [7]. O *Internet Protocol Support Profile* (IPSP) permite que dispositivos BLE descubram-se e comuniquem-se uns com os outros para a troca de pacotes IP.

A comunicação entre os dispositivos é feita através da troca de pacotes IPv6 sobre o BLE, onde esses pacotes são transmitidos por *LE-Frames*. A transmissão dos pacotes IPv6 não faz parte do perfil IPSP, e sim é definido pelo IETF (*Internet Engineering Task Force*) através da RFC 7668 (*Request For Comments*) [8]. A Figura 2.4 apresenta a pilha de protocolos utilizada pelo IPSP e pelo protocolo IPv6 sobre o BLE. É importante notar que os protocolos GATT e ATT são utilizados apenas para a descoberta do serviço IPSP. Após essa descoberta, a troca de dados é realizada através de um canal orientado a conexão no L2CAP BLE como descrito em seções anteriores.

O IPSP define dois modos de funcionamento: O modo Nó e o modo Roteador. O modo Roteador é implementado por dispositivos que realizam o roteamento de pacotes IPv6. O modo Nó é utilizado por dispositivos que criam e consomem pacotes IPv6. Um dispositivo em modo Nó implementa um servidor GATT, o qual faz o anúncio do serviço de suporte ao IP (IPSS).

Após esse anúncio, um dispositivo Roteador é capaz de descobrir um dispositivo Nó, e iniciar uma conexão com o mesmo, criando um canal orientado a conexão utilizando o modo de controle de fluxo de dados baseado em créditos. A partir desse ponto, *LE-Frames* são trocados entre o Roteador e o Nó, onde esses *LE-Frames* carregam em seu conteúdo pacotes

[1] https://www.bluetooth.org/en-us/specification/adopted-specifications

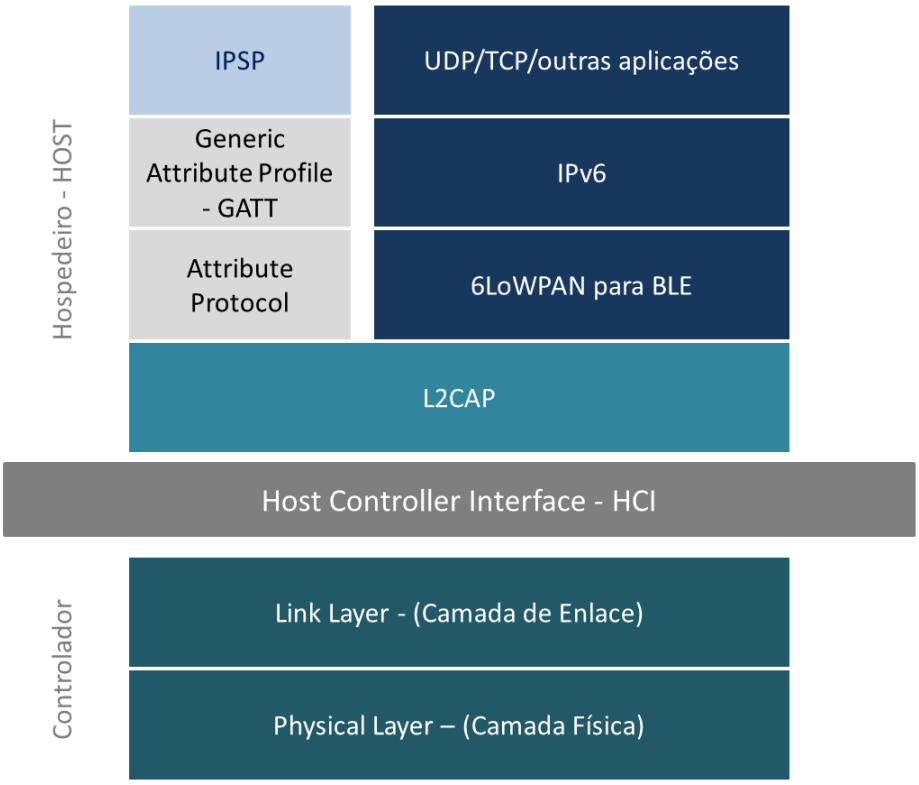


Figura 2.4: Pilha de protocolos para o suporte ao IPv6 no BLE.

IPv6.

Em relação a troca de pacotes IPv6 sobre BLE, a RFC 7668 define parâmetros de adaptação semelhantes ao do padrão 6LoWPAN [42]. As principais diferenças aparecem no que diz respeito a procedimentos de fragmentação, os quais são utilizados os procedimentos do BLE. Além disso, o BLE apenas suportar redes em formato estrela (*piconet*) ao contrário do IEEE 802.15.4, que é a tecnologia de transporte que o 6LoWPAN considera como base. Dentre as regras presentes na RFC 7668, destacam-se os procedimentos que devem ser adotados para a autoconfiguração de endereços, descoberta de vizinhos, compressão de *headers* e um modelo de enlace lógico (*link model*) para o IPv6, o qual deve considerar regras como o tamanho máximo de MTU (*Maximum Transmission Unit*) a ser utilizado.

## 2.2 Protocolos para a Internet das Coisas

O termo "Internet das Coisas" (*Internet of Things* - IoT) foi criado para descrever o fenômeno de conexão de diversos dispositivos com a Internet. Com o IoT, vários desafios e oportuni-

dades foram criados, desde a possibilidade da criação de novos serviços na Internet, até o desenvolvimento de novos meios de comunicação para dispositivos embarcados.

Em relação a comunicação desses dispositivos com a Internet, novas tecnologias de transmissão sem fio de baixo consumo foram criadas, como o Bluetooth Low-Energy (BLE). Além disso, protocolos de comunicação mais simples estão sendo desenvolvidos e avaliados. Esses protocolos, em geral, apresentam características para simplificação de mensagens, de modo a facilitar o seu uso em dispositivos com poucos recursos computacionais, como sensores.

Dentre os protocolos disponíveis, alguns destacam-se por sua aplicabilidade em redes *Machine-to-Machine* (M2M) e seu modelo simples de comunicação. Um desses protocolos é o MQTT (*Message Queue Telemetry Transport*). O MQTT foi inicialmente criado pela IBM, e sua especificação é aberta [5]. O MQTT funciona em nível de aplicação, considerando o modelo OSI (*Open Systems Interconnection*), e utiliza o protocolo TCP/IP como meio de transporte. O protocolo é baseado em um modelo *publisher/subscriber*, onde clientes criam uma conexão com um *broker* para a troca de mensagens. O MQTT também oferece diferentes níveis de QoS, dando flexibilidade para os desenvolvedores de serviços em sua rede. Apesar de ser um padrão bastante adotado, a utilização do protocolo TCP/IP impõe uma sobrecarga desnecessária a algumas aplicações. Nesse sentido, outro protocolo apresenta-se como uma alternativa para aplicações mais simples, o *Constrained Application Protocol*.

### 2.2.1 CoAP - Constrained Application Protocol

O *Constrained Application Protocol* (CoAP) é um protocolo Web criado para atender requisitos de dispositivos e redes com poucos recursos disponíveis. Apesar de seu nome se assemelhar ao do padrão SOAP (*Simple Object Access Protocol*, o modelo de comunicação do CoAP é semelhante ao REST (*Representational State Transfer*. O CoAP, ao contrário do MQTT, funciona sobre o protocolo UDP, e seu modelo de comunicação segue um padrão cliente/servidor, portanto, apresentando características semelhantes ao HTTP (*Hypertext Transfer Protocol*. Com isso, métodos equivalentes ao GET, POST, PUT e DELETE estão presentes no CoAP. Essa semelhança de modelos possibilita o uso do CoAP em conjunto com o HTTP através de *proxies* HTTP-CoAP [43] [33]. Esse tipo de interação, entre o CoAP

e outros protocolos, potencializa o seu uso por dispositivos embarcados na Internet, viabilizando a Internet das Coisas. Além disto, apesar de ser transportado por pacotes UDP, o CoAP suporta o envio de mensagens com confirmação de entrega, fornecendo confiabilidade ao transporte de pacotes, o que é um requisito essencial quando tratando-se de dados de saúde. Com uso de mensagens com confirmação de entrega, o CoAP oferece a possibilidade de enviar respostas em modo *piggybacking*[2], ou seja, ao confirmar o recebimento de uma mensagens, um dispositivo CoAP pode retornar uma mensagem extra em conjunto com a resposta de confirmação de recebimento.

Outra característica importante do CoAP é a possibilidade de descoberta de serviços através de um diretório de recursos (*CoRE Resource Directory*). Em termos de segurança e privacidade, o CoAP pode utilizar o *Datagram Transport Layer Security* (DTLS) [44]. Além disso, a utilização do CoAP traz benefícios relativos ao consumo de energia em dispositivos com poucos recursos [45].

O CoAP também pode ser utilizado sobre diferentes tecnologias de transporte. Apesar do CoAP ser desenvolvido tendo em vista o meio físico IEEE 802.15.4, novas propostas estão sendo apresentadas para o uso do CoAP em conjunto com outras tecnologias, como o Bluetooth Low-Energy (BLE) [6] e o padrão de mensagens SMS[3]. Com isso, o uso do CoAP em conjunto com o BLE viabiliza um novo conjunto de aplicações e serviços, dado a adoção do Bluetooth por vários dispositivos móveis e eletrônicos.

## 2.3 Sistemas de Saúde Conectada

Considerando o domínio de Saúde Conectada, torna-se importante discutir como a indústria e a comunidade científica estão utilizando padrões e tecnologias de comunicação em prol da interoperabilidade. Vários grupos de trabalhos, incluindo o grupo ISO/IEEE 11073, apresentaram padrões e recomendações para diversos níveis da cadeia de comunicação relativa a dados de saúde. Em especial, a associação *Continua Health Alliance* [4] desenvolveu recomendações para intercomunicação de sistemas e dispositivos de saúde na Internet

[2]piggybacking é a funcionalidade que permite que informações extras "peguem carona" em um pacote com outra finalidade.

[3]Mais detalhes em http://tools.ietf.org/wg/core/

[4]http://www.continuaalliance.org

[11]. Essas recomendações, chamadas de *Continua Design Guidelines (CDG)* foram adotadas como recomendação ITU-T H.810 [46]. O CDG apresenta uma arquitetura de referência onde interfaces são definidas para os diferentes níveis de comunicação, como ilustrado na Figura 2.5, e descritas a seguir:

- TAN-IF é a interface onde os DPS estão a um raio muito próximo de um agregador de Saúde (*Application Host Device* - AHD) em uma rede TAN (*Touch Area Network*), onde dados são trocados através de toque entre os dispositivos;
- PAN-IF é a interface onde os DPS estão a um raio próximo ao AHD do usuário, portanto, em uma rede pessoal PAN (*Personal Area Network*), como uma uma rede Bluetooth;
- LAN-IF é a interface entre o DPS e o AHD, onde ambos se comunicam a partir de uma rede local LAN (*Local Area Network*);
- WAN-IF é a interface entre o AHD e serviços na Internet através de uma rede WAN (*Wide Area Network*);
- HRN-IF é a interface entre um serviço de saúde WAN e serviços de uma *Health Record Network* (HRN), ou seja, serviços de armazenamento de dados de saúde.

Em termos de tecnologias de transmissão, o CDG define o *Near-Field Communication* (NFC)[5] como tecnologia TAN, o Bluetooth e o Bluetooth Low-Energy[9] como tecnologias PAN e o ZigBee[6] como LAN. Nesses níveis, o protocolo ISO/IEEE 11073 é utilizado como base para a comunicação entre os DPS e os agregadores.

Nas interfaces restantes, são utilizados perfis disponibilizados pelo *Integrating the Healthcare Enterprise (IHE)*[7][47]. Esse perfis, de maneira geral, fazem uso dos padrões disponibilizados pelo *Health Level 7 (HL7)* [8], os quais compreendem os padrões de maior adoção para a troca, gerenciamento, e integração de informações relativas a saúde em diferentes níveis, desde o nível clínico ao administrativo. Apesar de bastante amplo, o HL7 não define

[5] http://www.nfc-forum.com
[6] http://www.zigbee.org
[7] http://www.ihe.net
[8] http://www.hl7.org

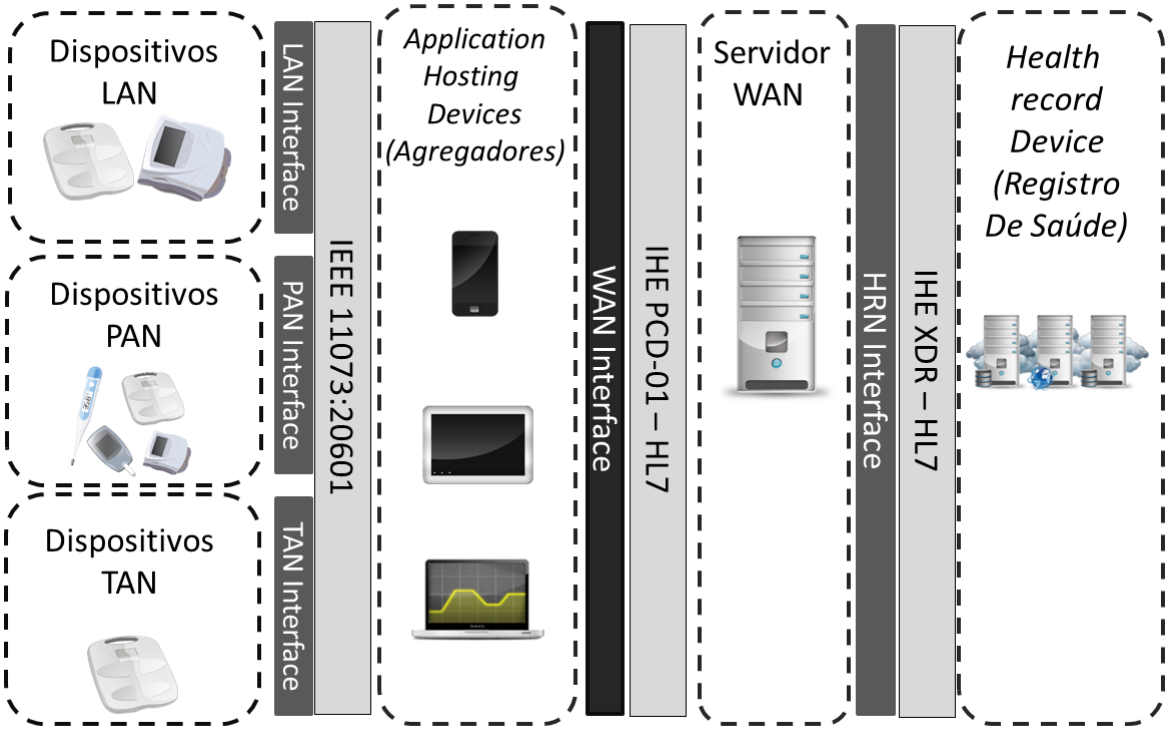


Figura 2.5: Arquitetura de Referência do Continua Health Alliance.

mensagens apropriadas ao uso por DPS, portanto, por isso da adoção do ISO/IEEE 11073 nos primeiros níveis de rede pelo CDG.

Considerando o contexto desse trabalho, o CDG apresenta uma arquitetura de referência para sistemas de Saúde Conectada e MRP. Entretanto, o CDG ainda considera o uso de agregadores de dados de saúde, os quais podem ser *smartphones* ou *tablets* por exemplo, para a coleta de dados dos DPS, adaptação do formato da informação e encaminhamento para a Internet. Portanto, o CDG não apresenta recomendações que permitam que os DPS compartilhem dados diretamente com serviços na Internet. Nesse sentido, na próxima seção apresenta-se uma classificação de DPS em relação ao seu modo de compartilhamento de dados com a Internet.

### 2.3.1 Dispositivos Pessoais de Saúde

Considerando um sistema de Saúde Conectada, o primeiro passo para seu funcionamento é a coleta e compartilhamento automatizado dessas informações utilizando interfaces de

comunicação como Bluetooth, ZigBee ou USB. Após a coleta, a informação deve ser enviada para a nuvem, e neste ponto podem-se classificar o DPS por seu modo de envio de informação:

- *DPS preparado para a Internet*. São dispositivos que geram a informação de saúde preparada para Internet, ou seja, já encapsulam a informação em um formato IP, de modo que não é necessária qualquer alteração para o tráfego de dados na Internet;
- *DPS dependente de Gateway ou Agregador de Dados de Saúde*. São dispositivos que geram informações de saúde e as compartilham utilizando um Gateway coletor, ou um agregador, o qual encapsula e transforma a informação para que esta seja enviada para a Internet.

A maioria dos DPS atuais disponíveis no mercado depende de Gateways de Dados de Saúde para enviar a informação para a Internet, como ilustrado na Figura 2.6. É importante perceber que os dados de saúde podem ser alterados durante o encapsulamento de dados no Gateway. Essa operação pode gerar perda ou alteração semântica dos dados, o que do ponto de vista de dados de saúde não é tolerado.

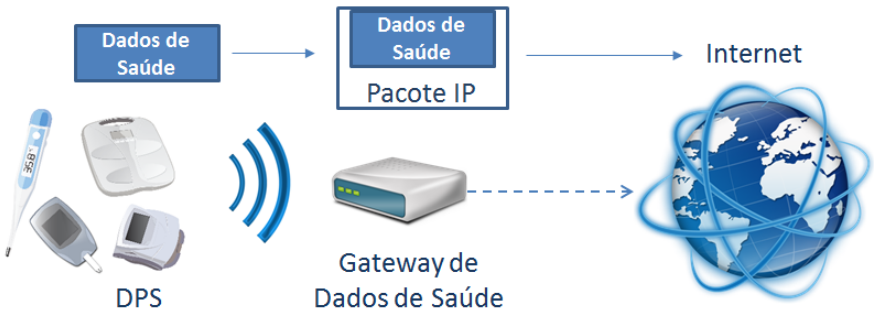


Figura 2.6: Comunicação entre um DPS e a Internet através de um agregador de dados de saúde.

Considerando DPS preparados para a Internet, o transporte dos dados é mais simples, como ilustrado na Figura 2.7. A informação é gerada e compartilhada em um formato pronto para tráfego na Internet, portanto, os dados apenas vão trafegar por Gateways de Internet, os quais não alteram a informação, reduzindo a probabilidade de erros na manipulação dos dados.

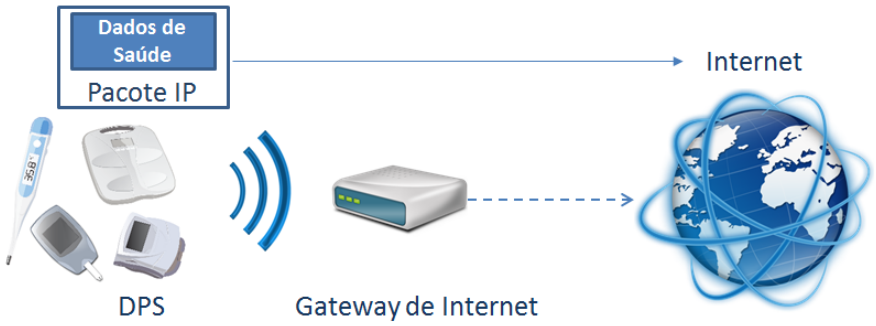


Figura 2.7: Comunicação entre um DPS e a Internet através de um Gateway de Internet.

Com essa definição de tipos, pode-se avaliar a estrutura interna de um DPS através de sua modularização em três módulos:

- o *Módulo de Sensoriamento*, o qual é responsável pela coleta das informações do meio físico através de sensores, e do envio para o módulo de controle;
- o *Módulo de Controle*, o qual é responsável pela manipulação e processamento dos dados recebidos dos sensores;
- o *Módulo de Comunicação*, o qual é responsável pelo envio e compartilhamento das informações de saúde coletadas pelo DPS.

No decorrer desse trabalho, abordagens e soluções para os *Módulos de Controle e Comunicação* serão propostos.

### 2.3.2 O Padrão ISO/IEEE 11073 para Dados de Saúde

Essa seção apresenta uma revisão simples sobre o padrão ISO/IEEE 11073. Dois tipos de dispositivos são definidos pelo ISO/IEEE 11073: agentes e agregadores. Agentes são produtores de dados, tipicamente dispositivos sensores como um DPS. Agregadores, por sua vez, são os coletores de dados. A conexão entre os dispositivos pode ocorrer em qualquer direção, entretanto, normalmente, o agente tem a iniciativa de conexão, pois o mesmo tem ciência de quando os dados dos sensores estão disponíveis, por exemplo, quando um paciente faz uma medição de glicose utilizando um DPS. O ISO/IEEE 11073 tem como base o requisito de que agentes são dispositivos sensores com poucos recursos computacionais e com limitações

de bateria, enquanto um agregador é tipicamente um dispositivo com mais recursos computacionais e está conectado a uma fonte de energia maior. Portanto, a maior complexidade computacional do ISO/IEEE 11073 está no agregador.

O documento base do protocolo é o ISO/IEEE 11073:20601 [12]. Além deste, entretanto, existem documentos que definem especializações para dispositivos. Por exemplo, o documento IEEE 11073:10404 define a especialização para um dispositivo oxímetro de pulso. Estas especializações definem como informações específicas destes dispositivos são transportadas pelo ISO/IEEE 11073. Também é definida que tipo de informação o dispositivo suporta, por exemplo, quais dados um glicosímetro deve salvar internamente.

Em especial, relativo ao objetivo deste trabalho, o ISO/IEEE 11073 é um protocolo independente de meio de transporte. Portanto, dados ISO/IEEE 11073 podem ser transportados por praticamente qualquer tecnologia de transmissão baseada em pacotes, como TCP/IP, Bluetooth ou USB. Vários DPS disponíveis no mercado fazem uso do Bluetooth HDP (*Bluetooth Health Device Profile*) como tecnologia de transporte. Outros dispositivos, por exemplo, fazem uso do perfil USB PHDC (*Personal Health Device Class*). Em ambos os casos esses perfis oferecem meios para o transporte de dados de saúde definidos pelo ISO/IEEE 11073.

A especificação do protocolo ISO/IEEE 11073 é definida utilizando a linguagem ASN.1 e os dados são trafegados entre os dispositivos através de APDUs (*Application Protocol Data Units*) [13]. Em uma camada superior é definido um DIM (*Domain Information Model*) [48], o qual define uma estrutura de dados para um agente específico. Um dispositivo agente instancia um conjunto de classes ISO/IEEE 11073, as quais têm atributos que definem medições, unidades, etc. Todo agente tem um objeto MDS (*Medical Device System*). Um objeto MDS contem atributos com informações do fabricante, especializações do agente, ID do sistema, entre outros. Atributos MDS podem ser obtidos e alterados através de operações de *Get/Set*. Um agregador, normalmente, coleta informações sobre o MDS do agente via relatórios de eventos e configurações.

Como a maioria dos protocolos, o fluxo de controle do ISO/IEEE 11073 é governado por uma máquina de estados. O diagrama da Figura 2.8 apresenta uma representação simplificada dessa máquina de estados. Dois estados principais são definidos, *Disconnected e Connected*. Quando no estado *Connected*, agente e agregador devem iniciar um procedimento de *Association*, passando pelos os estados de *Associating*, *Associated*, *Disassociating*

e *Unassociated*. Durante o estado de *Associated*, o agente e o agregador iniciam a troca de informações de configuração, para então entrar em operação (*Operating*), onde eles realmente trocam eventos com medições.

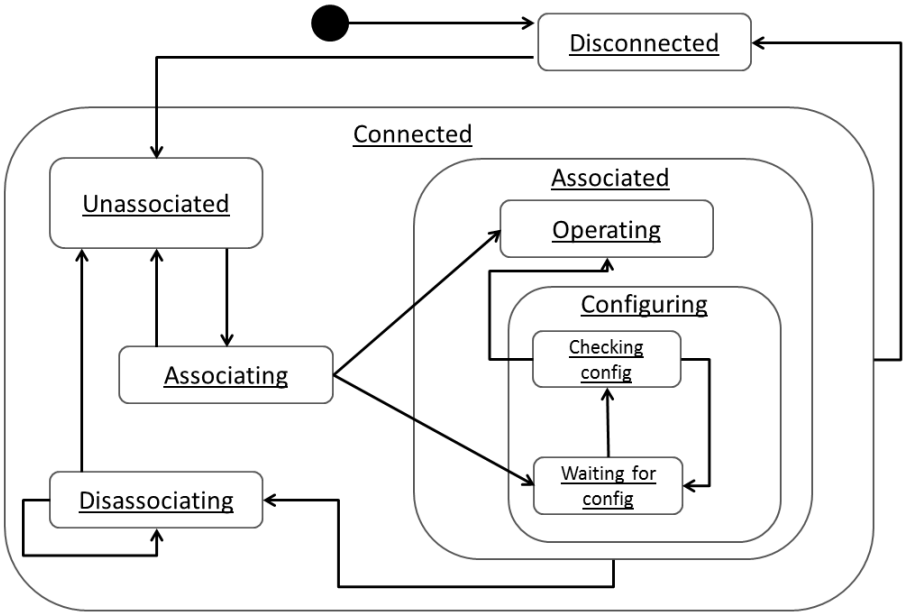


Figura 2.8: Máquina de Estados do ISO/IEEE 11073.

Após o procedimento de *Association*, mas antes de entrar em operação, o agente envia sua configuração ao agregador através de um APDU de evento. A configuração descreve detalhes do agente, como quantos objetos MDS existem e quais são seus atributos. Com essa informação, o agregador é capaz de interpretar eventos futuros do agente e, portanto, interpretar medições transmitidas pelo mesmo. Com isso, o agregador não precisa ter conhecimento prévio sobre o agente, e pode aprender detalhes do mesmo na fase de configuração.

## 2.4 Considerações Finais do Capítulo

Neste capítulo foram apresentados conceitos e tecnologias utilizadas no desenvolvimento do trabalho proposto neste documento. Os principais protocolos utilizados e suas descrições foram apresentados, tendo como alvo a tecnologia Bluetooth Low-Energy, e seu novo mecanismo de controle de fluxo baseado em créditos, o qual é utilizado para a transmissão de dados IP. É importante observar que, apesar de propor um esquema para o controle de fluxo

de dados, a especificação Bluetooth não apresenta uma política para a aplicação desse controle. Com isso, apesar de ser possível criar canais orientados a conexão em uma rede BLE, esses canais tem por padrão características de QoS de *best-effort*. Ou seja, não é possível diferenciar canais BLE de modo a garantir os requisitos de QoS necessários para diversos fins, como para sistemas de Monitoramento Remoto de Pacientes.

Por fim, em relação a Internet das Coisas e o caso de uso desse trabalho, foram apresentados detalhes sobre os motivos da escolha do protocolo CoAP, e sobre o funcionamento de sistemas de Monitoramento Remoto de Pacientes e seus protocolos.

# Capítulo 3

# Trabalhos Relacionados

Neste capítulo são apresentados trabalhos relacionados ao tema da tese apresentada nesse documento. Para a avaliação desses trabalhos, os mesmos foram divididos em grupos. Inicialmente realiza-se uma análise crítica sobre os trabalhos realizados para avaliação do Bluetooth Low-Energy e seus diversos usos. Em seguida, são apresentados trabalhos relacionados a modelos e sistemas de priorização de tráfego baseado em informações de contexto em redes PAN e BAN, os quais podem ser comparados com o trabalho apresentado nessa tese. Por fim, são apresentados alguns exemplos de sistemas de monitoramento de saúde voltados para a Internet, tendo como objetivo realizar uma análise comparativa com a solução de Monitoramento Remoto de Pacientes apresentado nesse trabalho.

## 3.1 Avaliação do Bluetooth Low Energy e o seu uso com o IPv6

Alguns trabalhos foram realizados para avaliar o Bluetooth Low-Energy em seus diversos aspectos. Em relação a aplicação do BLE destacam-se diversos trabalhos com a integração de sensores, como o trabalho de Touati et al. [49] o qual realiza uma avaliação experimental do uso de sensores de eletrocardiograma em uma rede sem fio corporal. Outro trabalho interessante, e com uma avaliação interessante foi apresentado por Bronzi et. al [50], o qual realizou experimentos com o BLE em um ambiente de comunicação veicular. Em ambos os trabalhos, os resultados mostraram a viabilidade e potencial técnico do BLE em diferentes

situações.

Considerando mais especificamente avaliações de desempenho e funcionamento, vários trabalhos realizaram trabalhos sobre o BLE. Do ponto de vista de modelagem e análise, os trabalhos de Liu et. at [51] criaram modelos analíticos os quais mostraram o impacto do procedimento de descoberta de vizinhos em redes BLE. Esses resultados são importantes, pois mostram que apesar de ser um procedimento simples, ele ainda apresenta um pequeno impacto na rede como um todo. O trabalho de Gomez et. al [52] apresentou um modelo analítico sobre a taxa de transmissão máxima de um enlace BLE. Os resultados desse modelo são apresentados na Figura 3.1. É interessante observar que em um modelo perfeito, sem taxa de erros de bits, a taxa máxima de transmissão em nível de aplicação em um enlace BLE é de 236.7 Kbps. Entretanto, considerando uma taxa de erro de bits (BER) de entre $2 \times 10^{-4}$ e $10^{-3}$, a qual é a taxa de sensibilidade do receptor definida pelo padrão BLE, e considerando um *connInterval* entre 30ms e 300ms, os quais são valores mais comumente utilizados por controladores BLE, é possível observar que a taxa de transmissão não ultrapassa 75 Kbps.

Considerando outros trabalhos de avaliação experimental, como o realizado em Mackensen et. al [53] e o realizado posteriormente por Gomez et. al [14], é possível observar que a em ambientes reais a taxa máxima de um enlace BLE em nível de aplicação não ultrapassa 58,48 Kbps a depender do controlador de hardware do BLE. Essa limitação vem de configurações do hardware, o qual tem limites no número de eventos que podem ser trocados durante um evento de conexão, assim, limitando a taxa de transmissão total.

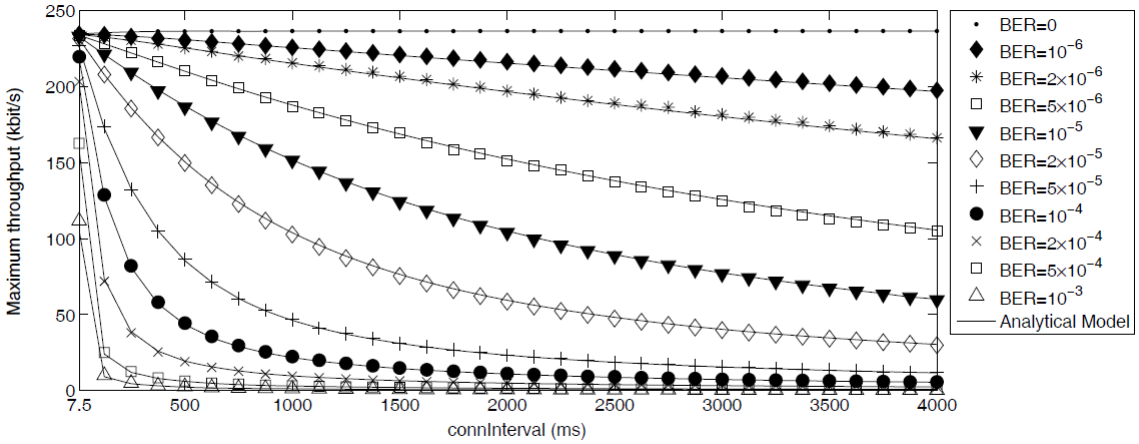


Figura 3.1: Resultados da taxa máxima de um enlace BLE apresentados no modelo analítico de Gomez et al.

Por fim, apesar de modelos analíticos terem sido realizados, poucos trabalhos apresentam ferramentas para simulação de redes BLE. Em especial, apenas o trabalho realizado por Mikhaylov et al. [54] apresentou alguns detalhes sobre um simulador em desenvolvimento, o qual utiliza o OMNet++ como base. Entretanto, esse trabalho não foi colocado a disposição do público até o momento.

Em relação ao seu consumo de energia, os trabalhos de avaliação de Gomez et al. [14], Nieminen et al. [6] e Siekkinen et al. [15], mostraram a partir de experimentos a viabilidade e o baixo consumo do BLE em diferentes situações. Observando os resultados de Siekkinen et al. é possível observar que mesmo realizando uma comparação com a tecnologia ZigBee/802.15.4, o BLE se mostrou bastante eficiente em termos da troca de dados por energia consumida.

Recentemente, a utilização de comunicação IPv6 sobre o BLE também vem sido avaliada por diversos trabalhos. Em especial, o trabalho de Nieminen et. al [6] apresenta uma visão geral do uso de BLE na Internet a partir de experimentos. O trabalho de Chawathaworncharoen et al. [55] realizou uma avaliação experimental do uso de 6LoWPAN sobre BLE, mostrando suas vantagens em relação ao consumo o comparando com conexões Wi-Fi. Este trabalho também apontou limitações das implementações atuais do 6LoWPAN sobre BLE. Outros trabalhos, como os apresentados em [56] e [57] apresentaram protótipos iniciais de comunicação IPv6 sobre BLE em estágios iniciais. Por fim, em relação ao uso do IPv6 sobre BLE, trabalhos recentes exploraram aplicações e usos diversos para esse novo modo de comunicação. Em Yim et al. [58] foi desenvolvido um serviço de *stream* de dados acústicos utilizando IP6 sobre BLE. Já no trabalho de Kim et al. [59] foi proposto um modelo para a criação de redes *mesh* utilizando BLE, permitindo assim novos tipos de aplicações.

Em comum, nenhum trabalho realizado sobre o BLE explorou aspectos relativos ao controle de fluxo de dados em seus Gateways (nós *masters*). Também, até o momento de escrita desse documento, não foram encontrados trabalhos que realizassem um estudo sobre o uso do controle de fluxo baseado em créditos, o qual foi recentemente introduzido na camada L2CAP como apresentado no Capítulo 2.

## 3.2 Contexto e Controle de Fluxo de Dados para Redes PAN e BAN

Considerando um cenário onde aplicações e serviços enviam informações de contexto para melhoria na camada de rede, a utilização de informação de contexto para adaptação de sistemas e redes sem fio é abordada em vários trabalhos. Alguns trabalhos de revisão recentes apresentam um resumo sobre as principais abordagens para utilização desse tipo de informação. Trabalhos como os apresentados em [60] e [61] detalham aspectos que devem ser considerados na utilização de contexto em redes sem fio e na Internet das Coisas. São apresentadas classificações para tipos de contexto, de como a informação de contexto pode ser processada, e de como essa informação pode ser utilizada no sistema ou rede em questão.

Seguindo essa linha de utilização de contexto para adaptação de redes móveis, outros trabalhos apresentam resumos e avaliações de como esse tipo de informação, como os trabalhos apresentados em [61] e [62]. Considerando redes PAN e BAN, esses trabalhos apresentam dois níveis de utilização de informação de contexto em sistemas e redes:

- Em nível de aplicação através de *middlewares* diversos, como os apresentados anteriormente.
- Em nível da camada de controle de acesso ao meio (MAC) e enlace.

Em nível de aplicação, diversos trabalhos utilizam informação de contexto para oferecer melhores serviços ao usuário final. Fazendo um paralelo ao modelo dinâmico de adaptação do fluxo de dados proposto nesse trabalho de tese, o trabalho apresentado por El Mougy et al. [63] apresenta um arcabouço colaborativo e distribuído que faz uso de um mecanismo multicamada para o compartilhamento de informação de contexto e recursos entre diferentes aplicações. O principal objetivo desse trabalho foi desenvolver um modelo onde recursos e aplicações possam ser compartilhados entre diferentes dispositivos em uma rede M2M. Apesar de propor um modelo onde recursos são distribuídos entre diferentes dispositivos, a abordagem de utilização de contexto pode ser utilizada para outros fins, como para o controle de fluxo de dados em camadas de rede inferiores.

Considerando os trabalhos relacionados ao uso de informação de contexto na camada MAC, várias abordagens foram propostas com aplicação em redes BAN e PAN. Dentre os

trabalhos propostos, e considerando os que apresentam um cenário de rede com topologia estrela, onde Gateways são utilizados como pontos centrais para os sensores, alguns trabalhos se destacam. O trabalho proposto por Yan et al. [64] apresenta uma proposta de controle de fluxo de dados baseado em informação de contexto, onde ciclos de tempo em uma rede TDMA são alterados a depender do tipo de informação que o sensor está enviando. O trabalho de Liu et al. [32] também apresenta uma abordagem semelhante, onde o controle é feito através do controle do fluxo TDMA. Em outro trabalho, apresentado por Rezvani et al. [65], informações relacionadas a qualidade do canal de transmissão são utilizadas para a adaptação de canais TDMA. Em comum, esses trabalhos apresentam o mesmo o problema descrito no Capítulo 1, onde dispositivos sensores compartilham o meio com outros dispositivos. Entretanto, eles adotam uma solução de ajuste temporal de *slots* de canal em um meio TDMA, portanto, atuando em um nível mais baixo na pilha de protocolos de rede. Esse tipo de abordagem dificulta sua implantação e validação, e com isso esses sistemas foram validados apenas através de simulações.

Outros trabalhos apresentam abordagens para criação de camadas MAC virtuais, as quais adaptam o fluxo de rede considerando o contexto de cada dispositivo sensor, como é o caso do BodyT2 [31] e do BodyQoS [30]. Apesar de apresentarem algoritmos diferentes, esses trabalhos tem em comum a utilização de um mecanismo de *polling* de dados. Ou seja, o nó central faz a requisição de dados para os sensores, portanto, impossibilitando que os mesmo sejam capazes de atuar autonomamente.

Considerando uma abordagem multicamada, onde a informação de contexto do usuário é utilizada para identificação da prioridade dos fluxos de dados, o trabalho apresentado por Carvalho et al. [66] apresenta uma abordagem onde é realizada uma adaptação no fluxo de dados a nível de transporte, diretamente no protocolo TCP/IP, como ilustrado na Figura 3.2. Apesar de adequado em alguns cenários, limitações na camada física e de enlace podem impor restrições nas adaptações aplicadas ao TCP/IP. Portanto, considerando redes com maiores restrições de banda de transmissão, como redes PAN BLE, deve-se considerar uma abordagem em níveis de rede mais baixos, como na camada de enlace.

Por fim, considerando a utilização de informação de contexto para adaptação de tráfego, outros trabalhos foram aplicados para diferentes topologias de redes, como redes *mesh*. Trabalhos como os descritos em [67], [68], [28], [27] e [69], apresentam soluções onde a

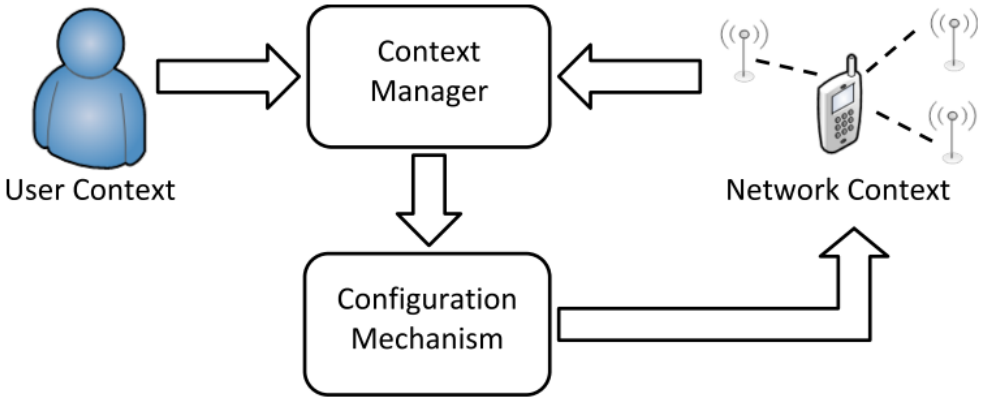


Figura 3.2: Diagrama com a Arquitetura de um sistema de adaptação de fluxo TCP/IP baseado em contexto apresentado por Carvalho et al.

adaptação de tráfego é feita entre outros pontos da rede, como por exemplo, entre o Gateway e o Serviço na Internet.

## 3.3 Sistemas de Monitoramento de Saúde para Internet

Com a evolução tecnológica e o novo paradigma da Internet das Coisas, novas oportunidades foram criadas e discutidas, como apresentado nos trabalhos de Aragues et al. [18] e [10]. Nessa seção, portanto, apresentam-se as principais e mais recentes iniciativas relacionadas a sistemas de Monitoramento Remoto de Pacientes (MRP) na Internet. Para cada sistema é realizada uma análise relacionada a abordagem utilizada para a coleta de dados dos dispositivos sensores.

O trabalho de Seeger et al. [22] apresenta um *middleware* para o desenvolvimento de aplicações de saúde para *smartphones*, o MyHealthAssistant. Esse *middleware* tem uma arquitetura de comunicação baseada em eventos, onde eventos originados de sensores de redes sem fio pessoais (ou corporais), enviam dados para um barramento de mensagem que o compartilha com aplicações no *smartphones*. Além dessas aplicações, são oferecidos serviços de monitoramento, segurança e privacidade. Em outro trabalho, Benharref et al [25] apresenta um arcabouço de serviços para o monitoramento em tempo real de pacientes. Esse arcabouço, chamado de SOCBeS, tem uma arquitetura orientada a servicços (*Service-*

*Oriented Architecture* - SOA), a qual objetiva o monitoramento de pacientes continuamente, e o envio de recomendações de maneira proativa. Uma abordagem semelhante foi apresentada por Fengou et al. [70], onde uma extensão da arquitetura ETSI/Parlay[1] é proposta para a criação de novos serviços de saúde na Internet. Por fim, o sistema GiraffPlus apresentado por Palumbo et al. [24] apresenta uma infraestrutura de rede sensores organizada por um *middleware* central. Esse *middleware* interage com sensores corporais e no ambiente com o objetivo de tomar ações de maneira proativa em prol do paciente. Todos os dados são coletados e enviados a um serviço central na Internet. Por fim, com o auxilio das informações dos sensores e do servidor central, o GiraffPlus pode se conectar com um robô remoto para intermediar a comunicação entre o paciente e um profissional remotamente. Entretanto, em nenhum desses trabalhos são abordados problemas relacionados a comunicação dos sensores e dispositivos que coletam os dados em suas redes locais.

Em uma linha mais relacionada a redes de sensores, o trabalho de Xiaonan et al. [26] propõe uma arquitetura para rede de sensores sem fio que pode ser utilizada para o monitoramento remoto de pacientes. A arquitetura desse sistema é ilustrada na Figura 3.3. Vários aspectos de MRP foram discutidos e avaliados nessa proposta de rede, sendo que uma de suas principais características é a utilização de uma rede *All-IP*, onde os sensores enviam dados diretamente para a Internet através do protocolo IPv6. Nesse sistema o IPv6 é utilizado para outros fins, como o de localização, por exemplo. Entretanto, esse modelo é apenas validado através de simulações, e aspectos relacionados a comunicação no primeiro nível de rede, entre o dispositivo sensor e o gateway, não são considerados.

Outro trabalho proposto por Jung et al. [71] apresenta uma solução M2M baseada em IPv6 onde dispositivos móveis são utilizados como Gateways. Um sistema real foi desenvolvido e experimentos avaliados, os quais mostraram a viabilidade do sistema, como ilustrado na Figura 3.4. Entretanto, apenas aspectos técnicos são abordados nesse trabalho. Uma avaliação do funcionamento desse sistema em um cenário mais amplo, onde diversos dispositivos compartilham a mesma rede, não é abordado.

[1] O Parlay foi transferido para o OMA: http://openmobilealliance.org/

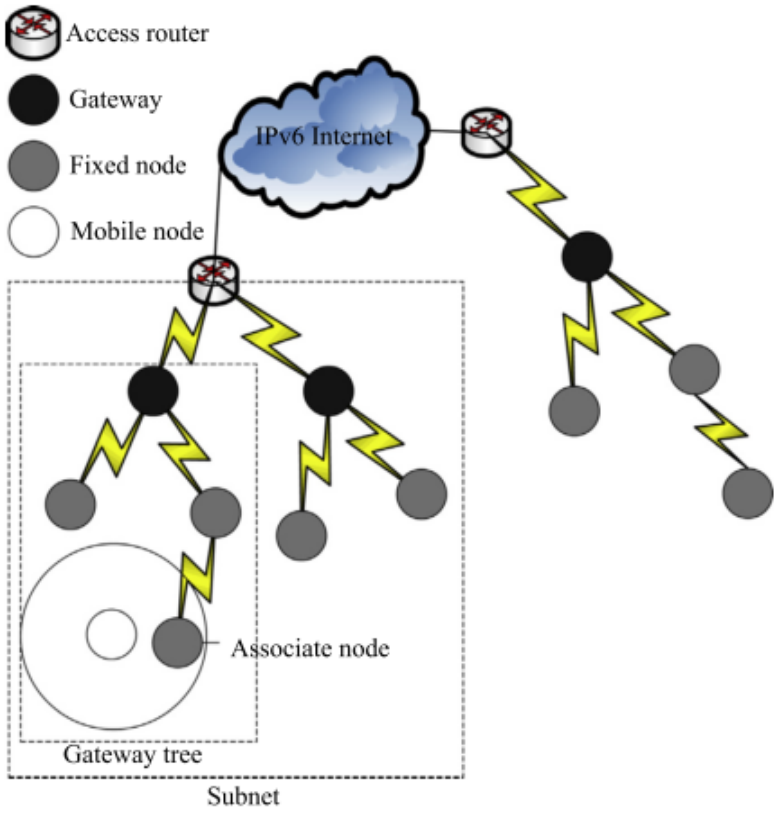


Figura 3.3: Diagrama com a arquitetura da rede de sensores sem fio All-IP.

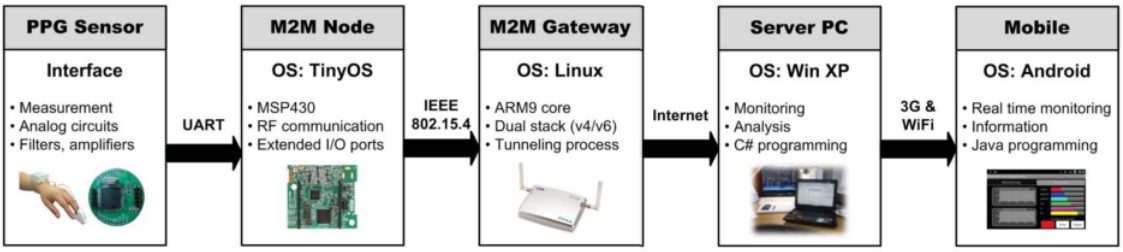


Figura 3.4: Diagrama com arquitetura de um sistema M2M para o MRP apresentado por Jung et al.

Uma arquitetura mais ampla foi proposta por Niyato et al. [29], onde serviços de MRP na Internet realizam de maneira proativa uma reserva da banda de transmissão para o recebimento de dados através de uma análise estatística do tráfego. Ao mesmo tempo, esse trabalho propõe um modelo de escalonamento de tráfego entre diferentes sensores na rede corporal baseado em uma pré-seleção, como ilustrado na Figura 3.5. Dados dos sensores são

classificados como críticos ou não, e um escalonador de tráfego faz uma decisão através de um processo de cadeia de Markov. Entretanto, apenas uma avaliação analítica foi realizada em conjunto com resultados de simulação. Também, no primeiro nível de rede, onde os sensores enviam dados a um agregador, não foram considerados aspectos relativos a redes sem fio.

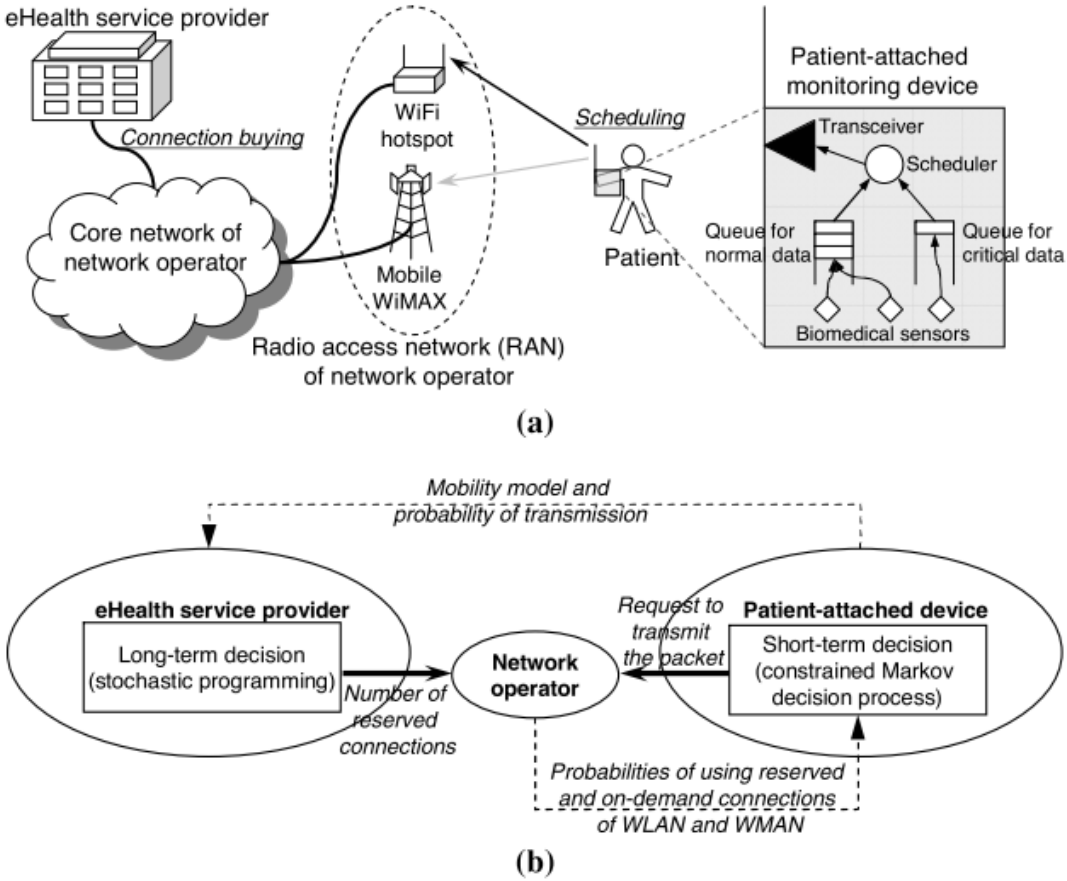


Figura 3.5: Diagrama com a arquitetura de um sistema MRP em redes heterogêneas apresentado por Niyato et al.

## 3.4 Considerações Finais do Capítulo

Com isso, dentre os trabalhos pesquisados na literatura, não foi encontrado uma proposta que apresente uma abordagem multicamada para avaliação de informação de contexto a partir da análise de tráfego em nível de aplicação, e que utilize essa informação para o controle

dinâmico do fluxo de dados em uma rede PAN em nível de enlace, e mais especificamente, em uma rede BLE.

Também, não foram encontrados trabalhos de avaliação do novo mecanismo de controle de fluxo baseado em créditos introduzido no Bluetooth 4.2, como também, trabalhos apresentando novos mecanismos de controle com esse mecanismo.

Em relação a adaptação dinâmica do controle de fluxo, características específicas de redes PAN e da camada de enlace do BLE não são consideradas nos trabalhos avaliados. Além disso, dentre os trabalhos analisados, sua avaliação e validação, em geral, é realizada através de simulações de redes IEEE 802.15.4 ou Wi-Fi, ou através de uma abordagem analítica, onde se considera que o contexto é aferido a priori, e não atualizado dinamicamente de maneira temporal.

Por fim, sobre a aplicação de um controle de fluxo adaptativo em um caso de uso de sistema de Monitoramento Remoto de Pacientes, apesar de propostas amplas terem sido apresentadas, essas não apresentam detalhes de como a aferição de contexto é realizada, ao contrário do que é apresentado nesse trabalho.

# Capítulo 4

# Arquitetura do Sistema de Monitoramento Remoto de Pacientes para a Internet das Coisas

Além dos requisitos e limitações de dispositivos embarcados, como pouca capacidade de processamento, alimentação através de baterias e interfaces de comunicação limitadas, Dispositivos Pessoais de Saúde (DPS) precisam ser confiáveis e seguros. Não apenas em relação a privacidade dos dados, mas também no que diz respeito à garantia na coleta e transmissão dos dados. Ou seja, DPS precisam garantir que o dado coletado pelo sensor de sinais vitais seja transmitido e entregue ao receptor final sem alteração. Além disso, requisitos de Qualidade de Serviço (QoS) devem ser considerados para o canal de comunicação. Por exemplo, a depender do canal de comunicação e suas características de transmissão, um DPS e o sistema de monitoramento podem ser utilizados para situações de emergência ou não [72].

Quando se faz uma análise desses dispositivos em conjunto com o novo paradigma da Internet das Coisas, novos requisitos aparecem. Na Internet, recursos e serviços estão disponíveis através de *Web Services*, e comumente parte da comunicação segue um modelo HTTP ou similar. Portanto, um requisito desejado para novos DPS com conectividade é a possibilidade de interação com serviços na Internet. Portanto, com o propósito de disponibilizar serviços de monitoramento de saúde para Internet das Coisas, nesse capítulo é apresentada uma arquitetura de referência baseada em padrões para DPS conectados.

## 4.1 Visão Geral

Como introduzido em capítulos anteriores, um Sistema de Monitoramento Remoto de Pacientes (SMRP) utiliza tecnologias de comunicação, com ou sem fio (*wireless*), para a coleta e compartilhamento automático de Informações Pessoais de Saúde (IPS). Considerando a classificação de Dispositivos Pessoais de Saúde (DPS) apresentada no Capítulo 2, uma arquitetura padronizada para o compartilhamento IPS foi definida neste trabalho. Nesta arquitetura, DPS enviam mensagens utilizando o padrão ISO/IEEE 11073 através de um modelo de comunicação REST, utilizando o protocolo CoAP como meio de transporte. Com esse modelo de comunicação, esses dispositivos são capazes de compartilhar dados em suas redes pessoais ou locais, como também diretamente com serviços na Internet. O CoAP foi escolhido dado seu modelo de comunicação REST, o qual facilita sua adoção e integração com outros serviços na Internet, além de oferecer a possibilidade de ser utilizado sobre diferentes meios de transporte, como o UDP ou até mesmo o SMS.

Dado que DPS baseados em Gateways estão amplamente disponíveis no mercado, alterar seu modo de comunicação e arquitetura não é uma opção. Consequentemente, torna-se necessário prover suporte a esses dispositivos legados em novos sistemas de monitoramento, com o propósito de manter compatibilidade com a infraestrutura já existente para o usuário final. Portanto, na arquitetura proposta, esses DPS legados são integrados ao sistema através de sistemas secundários, os quais fazem a integração dos mesmos com a infraestrutura existente através de outros modos de comunicação, como o UPnP (*Universal Plug and Play*[1]). Por exemplo, a Figura 4.1 ilustra um cenário onde dois fluxos de informação são apresentados. Em um primeiro fluxo, a informação pode ser compartilhada em redes locais através da utilização do UPnP. No segundo fluxo, a informação é compartilhada diretamente com a Internet através da utilização do protocolo CoAP e DPS preparados para a Internet.

Considerando a arquitetura proposta para a Internet das Coisas, nas próximas seções serão apresentados detalhes sobre o modelo de comunicação para DPS preparados para Internet.

[1] http://www.upnp.org

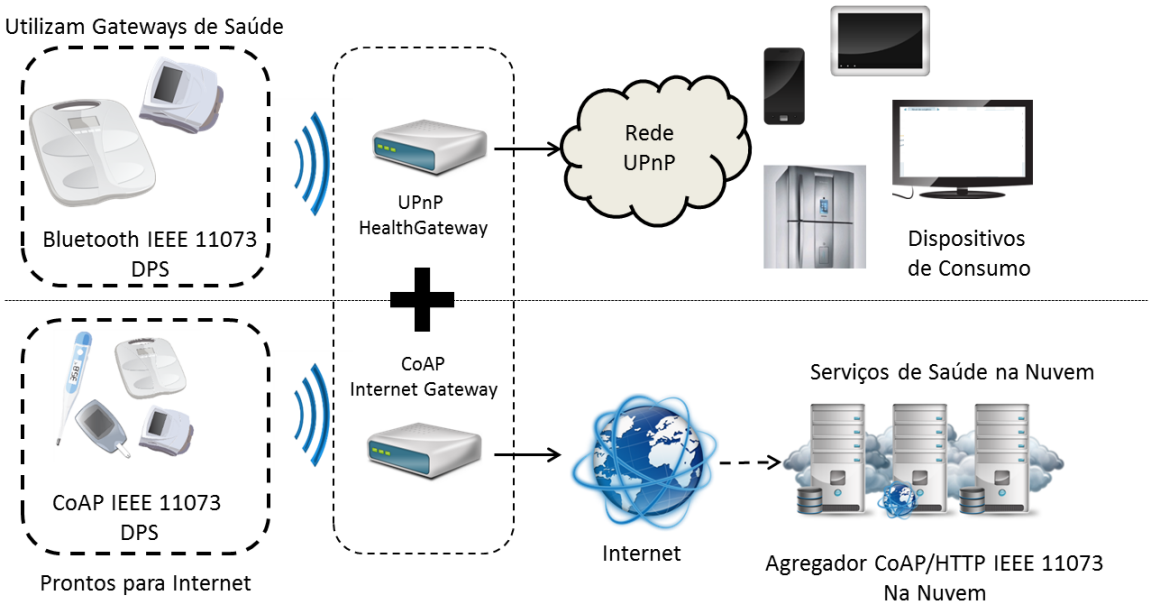


Figura 4.1: Diagrama com esquema de comunicação um sistema de referência ISO/IEEE 11073 com CoAP.

### 4.1.1 Adaptando o ISO/IEEE 11073 para o modelo de comunicação REST

O modelo de comunicação do protocolo CoAP é similar ao modelo do protocolo HTTP, como detalhado no Capítulo 2. Com isso, métodos similares ao GET, POST, PUT e DELETE estão disponíveis no CoAP. Entretanto, como também detalhado no Capítulo 2, o modelo de comunicação do ISO/IEEE 11073 é baseado em canais bidirecionais, onde agentes e agregadores trocam APDUs em canais com garantia de entrega e de modo assíncrono e confiável. Portanto, o primeiro passo para a integração do CoAP com o ISO/IEEE 11073 é fazer a adaptação e mapeamento do modelo de comunicação orientado à conexão do ISO/IEEE 11073 ao modelo cliente/servidor do CoAP baseado em REST.

No modelo de comunicação ISO/IEEE 11073, agentes ou agregadores podem iniciar a comunicação. Para manter essa característica de comunicação utilizando o CoAP, foi adotado o modelo onde agentes e agregadores são ao mesmo tempo clientes e servidores, como ilustrado na Figura 4.2. Na arquitetura proposta nesse trabalho foi considerado que agregadores ISO/IEEE 11073 devem ser obrigatoriamente servidores CoAP, e que agentes ISO/IEEE 11073 devem ser obrigatoriamente clientes CoAP, onde a comunicação sempre pode

ser iniciada pelo agente. Essa restrição se torna adequada e aceitável, pois, normalmente, os agentes iniciam a comunicação quando têm dados disponíveis para compartilhamento. Como uma característica adicional, agentes podem instanciar um servidor CoAP para compartilhar recursos observáveis através de eventos CoAP [73].

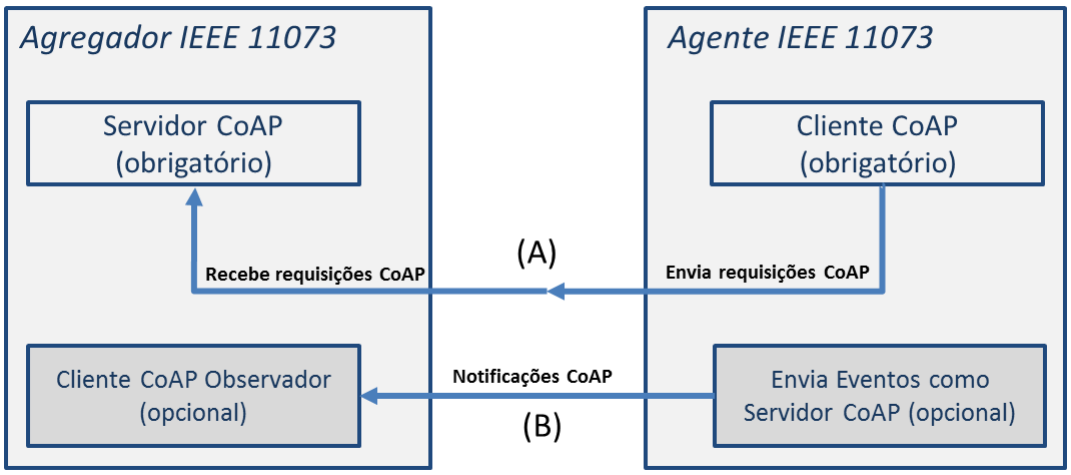


Figura 4.2: Mapeamento de entidades IEEE 11073 para o modelo CoAP.

Considerando que o conceito de canais ponto-a-ponto não existe no modelo REST, algumas considerações foram feitas de modo a manter o mesmo conjunto de estados do modelo ISO/IEEE 11073, como citado a seguir:

- quando um agente envia uma requisição de *Associação* ao agregador, ambos os dispositivos entram no estado *Associando* e *Conectado* automaticamente;
- os estados *Desassociado* e *Conectado* não existem;
- quando um dos dispositivos envia uma requisição de *Desassociação*, ou a *Associação* não é estabelecida, ambos os dispositivos vão para o estado de *Desconectado* automaticamente.

Para a transmissão de APDUs ISO/IEEE 11073, mensagens do tipo PUT são trocadas entre o agente CoAP (*11073Client*) e o agregador CoAP (*11073Server*) utilizando uma URL pré-definida. Apesar do ISO/IEEE 11073 suportar um número maior de mensagens e transações, apenas as seguintes transações foram consideradas nessa primeira versão da arquitetura:

- *Associação*: O *11073Client* envia uma mensagem de requisição de *Associação*, e o *11073Server* responde (no campo *body* da resposta da mensagem PUT) com a mensagem de resposta da *Associação*. Caso o *11073Server* em sua resposta reporte que não suporta a *Configuração* ISO/IEEE 11073 compartilhada pelo *11073Client*, este deverá enviar um *Informe de Configuração* na próxima mensagem PUT transmitida;

- *Informe de Configuração*: O *Informe de Configuração* apresenta detalhes sobre como os dados do agente *11073Client* são transmitidos dentro de um APDU. Desse modo, um agregador *11073Server* é capaz de extrair as informações desejadas de um APDU de *Evento*;

- *Requisição de Atributos*: Caso o *11073Server* deseje obter informações dos atributos ISO/IEEE 11073 do DPS, a requisição de atributos pode ser enviada na resposta da requisição de *Associação* utilizando a funcionalidade de *piggybacking* do CoAP;

- *Eventos*: Após a transação de *Associação*, ambos os dispositivos estarão no estado *Associado*, e o *11073Client* pode enviar eventos ao *11073Server*. É através de eventos que os DPS compartilham dados biométricos (como níveis de pressão arterial, batimentos cardíacos, etc). Para compartilhar dados de saúde, portanto, o DPS deve estar no estado *Associado*;

- *Desassociação*: Ao final, o *11073Client* envia uma requisição *11073Server*. Após a desassociação, o *11073Server* considera o *11073Client* como desconectado. Como método alternativo, um período de *timeout* pode ser adicionado a ambos os dispositivos para que ambos entrem no estado *Desconectado* após um período de tempo sem atividade.

Com esse mapeamento e definição de transações, o *11073Client* é capaz de enviar eventos ao *11073Server* utilizando mensagens CoAP e, ao mesmo tempo, seguir o modelo de comunicação ISO/IEEE 11073. A Figura 4.3 apresenta o fluxo de dados entre o *11073Client* e o *11073Server* considerando as transações descritas anteriormente.

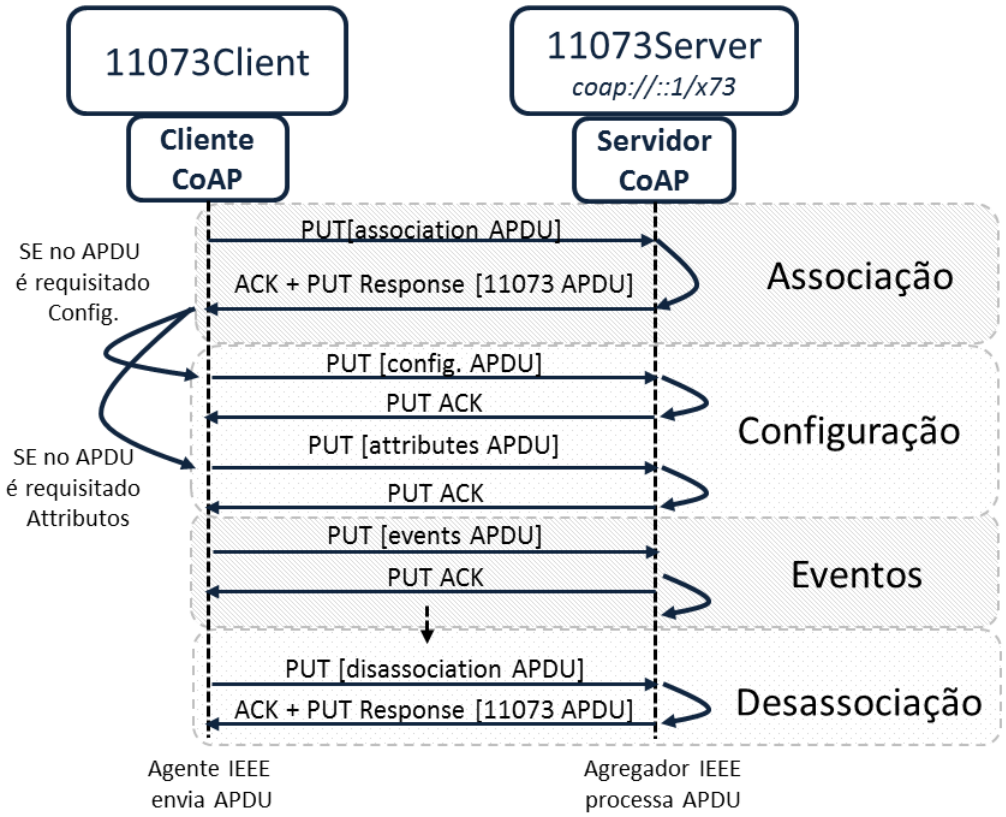


Figura 4.3: Fluxo de dados IEEE 11073 em um modelo cliente/servidor CoAP.

### 4.1.2 Desenvolvimento e Avaliação

Com o objetivo de avaliar a arquitetura proposta e seu desempenho em relação ao uso do protocolo CoAP, um sistema de referência foi desenvolvido utilizando ferramentas de software livre. Em especial, duas ferramentas foram amplamente utilizadas: A biblioteca ISO/IEEE 11073 Antidote [2] e a biblioteca libCoAP[3]. O Antidote oferece um conjunto de ferramentas e componentes de software que permitem o desenvolvimento de aplicações utilizando os padrões ISO/IEEE 11073. O Antidote tem uma arquitetura baseada em *plug-ins* que permite o suporte a novas tecnologias de transporte dinamicamente, como o Bluetooth HDP e o TCP/IP.

Com as ferramentas escolhidas, foram desenvolvidos dois componentes principais: os módulos CoAP ISO/IEEE 11073 cliente e servidor. Utilizando a arquitetura baseada em

[2]http://oss.signove.com

[3]http://sourceforge.net/projects/libcoap/

*plug-ins* do Antidote, foi desenvolvido um *plug-in* para o protocolo CoAP utilizando a biblioteca libCoAP, de modo que foram criadas duas entidades: o CoAP *11073Server* e *11073Client*. Ambos foram desenvolvidos e executados em um ambiente Linux, e uma especialização ISO/IEEE 11073 para oxímetro foi utilizada para os testes com o *11073Client*. A Figura 4.4(A) ilustra a arquitetura base para o CoAP cliente e servidor.

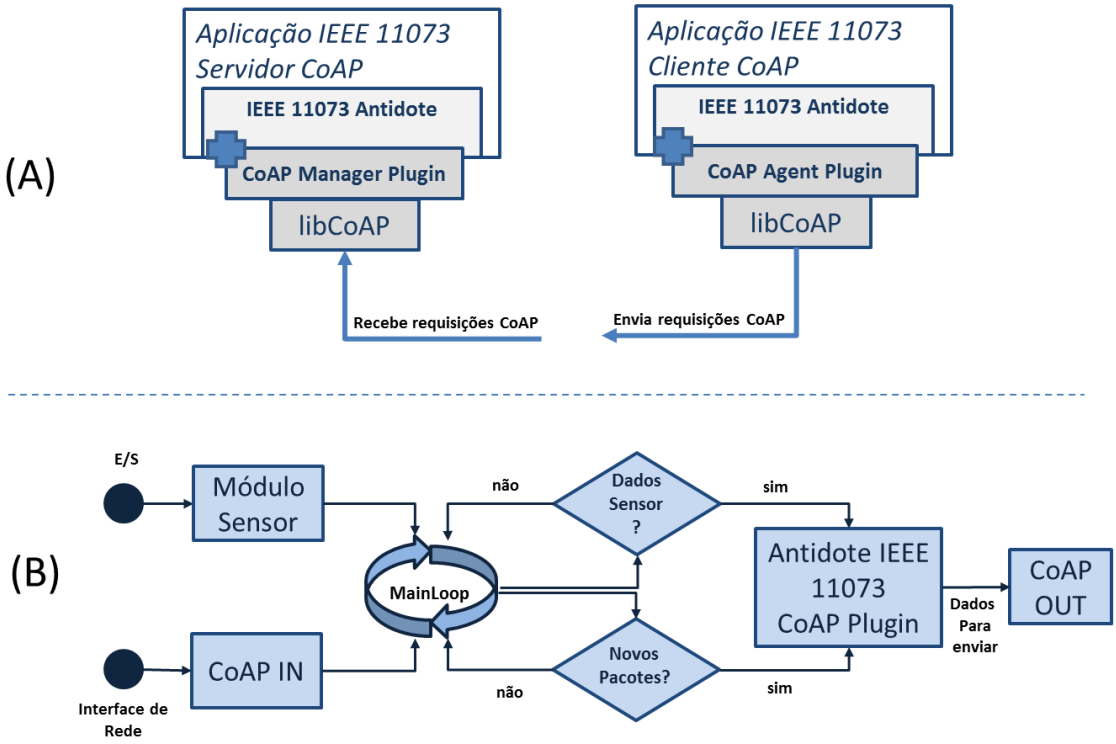


Figura 4.4: Diagrama com arquitetura interna de referência de um DPS CoAP.

A Figura 4.4(B) apresenta o fluxo de dados interno de um dispositivo embarcado com o *11073Client*. Com um escalonador central, dados são recebidos do módulo de sensoriamento e da interface de comunicação de rede. O módulo de sensoriamento pode receber dados de um sensor real ou de um módulo de simulação. Essa arquitetura base foi integrada em uma plataforma embarcada para a criação de um oxímetro real utilizando o ISO/IEEE 11073 e o CoAP. O dispositivo de referência é apresentado na Figura 4.5. No protótipo apresentado, uma interface de comunicação Wi-Fi foi utilizada para comunicação com o *11073Server*.

Posteriormente, esse sistema de referência foi integrado com outras duas soluções:

- uma solução UPnP desenvolvida para o compartilhamento de dados de saúde utilizando o ISO/IEEE 11073 [35];

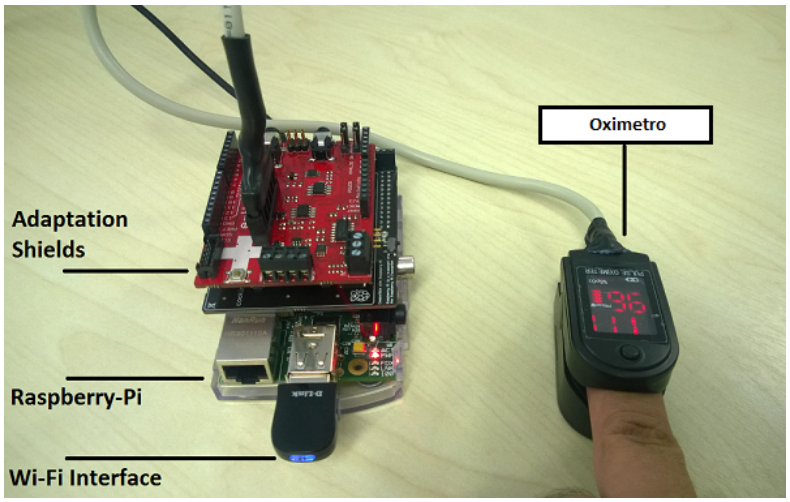


Figura 4.5: Protótipo de um Dispositivo Pessoal de Saúde com CoAP.

- uma plataforma de saúde conectada disponível comercialmente pela Signove Tecnologia, conhecida como SigHealth [4].

A partir do sistema de referência desenvolvido, alguns experimentos foram realizados para avaliação de desempenho e validação de funcionalidades.

### 4.1.3 Avaliação do Protocolo CoAP

Os testes realizados nessa etapa têm como objetivo avaliar o desempenho do modelo de comunicação REST adaptado ao ISO/IEEE 11073. Como referência foi utilizada uma solução do ISO/IEEE 11073 sobre *sockets* TCP/IP. *Sockets* foram escolhidos como modelo de referência dada sua ampla utilização na Internet, e por eles se adequarem aos requisitos de comunicação do ISO/IEEE 11073. Além de um *plug-in* CoAP, foi utilizado um *plug-in* TCP/IP em conjunto com a biblioteca Antidote. Durante os testes, o mesmo conjunto de transações ISO/IEEE 11073 foi utilizado entre o agente e o agregador. O objetivo desses testes é avaliar quantos pacotes e bytes são transmitidos ao total e, consequentemente, avaliar a carga de cada protocolo.

Duas transações foram comparadas:

- *Transação completa*: O *11073Client* inicia uma *Associação*, envia atributos, em seguida três medições através de *Eventos* e por fim desassocia. Nesse processo nove (9)

[4] http://health.signove.com

APDUs ISO/IEEE 11073 e 425 bytes são trocados em nível de aplicação;

- *Transação Simples*: O *11073Client* inicia uma *Associação*, em seguida envia uma medição através de um *Evento* e por fim desassocia. Nesse processo cinco (5) APDUs ISO/IEEE 11073 e 172 bytes são trocados em nível de aplicação.

A Tabela 4.1 apresenta os resultados comparativos entre os dois protocolos por tipo de transação.

Tabela 4.1: Comparação entre transações IEEE 11073.

| Protocolo de Transporte | Trans. Completa | Trans. Simples |
|---|---|---|
| CoAP REST | 12 pacotes<br>1295 bytes | 6 pacotes<br>599 bytes |
| TCP/IP Socket | 24 pacotes<br>2039 bytes | 16 pacotes<br>1244 bytes |

Como esperado, o modelo de comunicação REST utilizado pelo CoAP apresenta vantagens sobre a comunicação utilizando sockets TCP/IP. É importante notar que em uma transação simples o modelo CoAP envia até 50% menos pacotes que a solução TCP/IP. Esse é um resultado importante, dado que mostra claramente quanto pacotes a solução CoAP utiliza em relação ao TCP/IP, portanto, economizando recursos no DPS. A maior parte dessa sobrecarga vem da diferença entre o uso de UDP e TCP. Entretanto, além do uso de UDP, uma das principais razões para a diminuição do número de pacotes trocados vem do uso de mensagens *piggybacked* em cada pacote ACK do CoAP. Na solução CoAP/IEEE 11073, cada pacote ACK transporta a resposta da requisição ISO/IEEE 11073 anterior. Por exemplo, quando uma requisição PUT é enviada ao *11073Server*, no mesmo pacote de ACK do CoAP o *11073Client* receberá a resposta da requisição ISO/IEEE 11073.

### 4.1.4 Avaliação de Tráfego de Rede

Alguns testes foram realizados para avaliar algumas funcionalidades do CoAP, como o envio de mensagens com confirmação e a retransmissão de pacotes. Para a execução desses testes, um ambiente de simulação foi preparado em uma distribuição Linux. Foi utilizado a ferramenta de *Traffic Control (TC)* do Linux para a manipulação de pacotes diretamente na

interface de rede. Com essa ferramenta foi possível simular perda e atrasos na entrega de pacotes.

Nesse ambiente de testes, o principal objetivo foi avaliar o número total de transações ISO/IEEE 11073 completadas em uma rede WAN. Para simular esse tipo de rede, um canal foi modelado onde para cada pacote enviado pela interface de rede seria aplicada uma função de distribuição para o atraso no envio. Foi considerado que atrasos são modelados utilizando uma distribuição *Normal*. Foram considerados os seguintes parâmetros para a modelagem do canal:

$$\begin{aligned} Media &: \mu = 120ms \\ DesvioPadrao &: \sigma = 40ms \end{aligned} \tag{4.1}$$

Esse modelo descreve que 95% dos pacotes enviados tem um atraso de $\mu \pm 1,96 \times \sigma$. O valor de $\mu$=120 ms foi escolhido após experimentos onde foi observada a soma da média do tempo de envio de uma pacote ISO/IEEE 11073 em duas redes: Uma rede Bluetooth ($\mu$=20 ms) e na Internet ($\mu$=100 ms).

O gráfico apresentado na Figura 4.6 apresenta o tempo de envio de varias transações em segundos. O gráfico compara a solução CoAP com a solução TCP/IP descrita anteriormente. Os resultados mostram que o tempo médio de uma transação CoAP é menor que a solução TCP/IP. É importante observar que o tempo de uma transação TCP/IP é maior por causa dos procedimentos de conexão dos sockets. Entretanto, mesmo removendo esses procedimentos de conexão (aproximadamente 500 ms), a média de tempo do CoAP é menor.

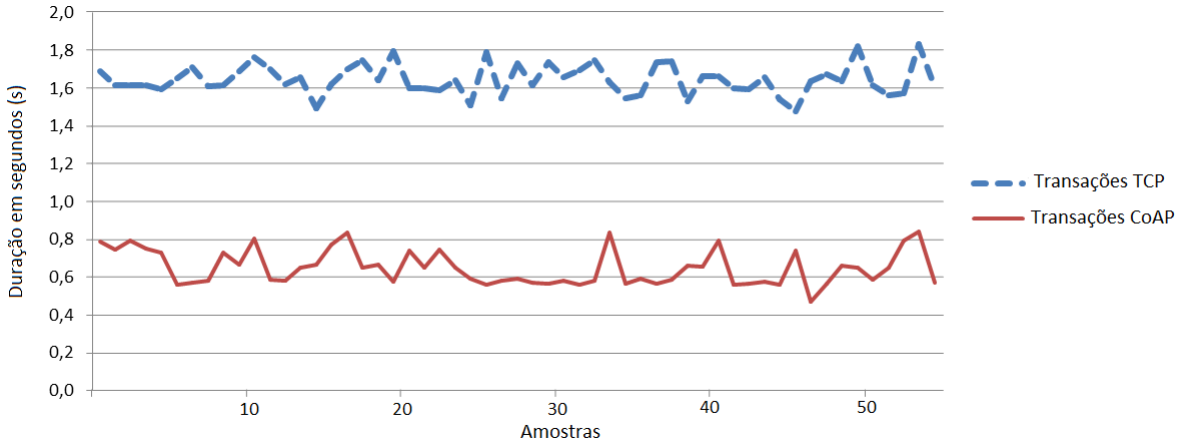


Figura 4.6: Comparação entre transações IEEE 11073 utilizando o CoAP e TCP/IP.

O próximo teste tem como objetivo avaliar a característica de retransmissão do CoAP. Para tanto o canal modelado anteriormente foi alterado para adicionar uma taxa de perda de pacotes de 10%. Esse valor foi escolhido para forçar a perda de pacotes com uma frequência maior. O gráfico da Figura 4.7 apresenta o número total de pacotes trocados para cada transação simples.

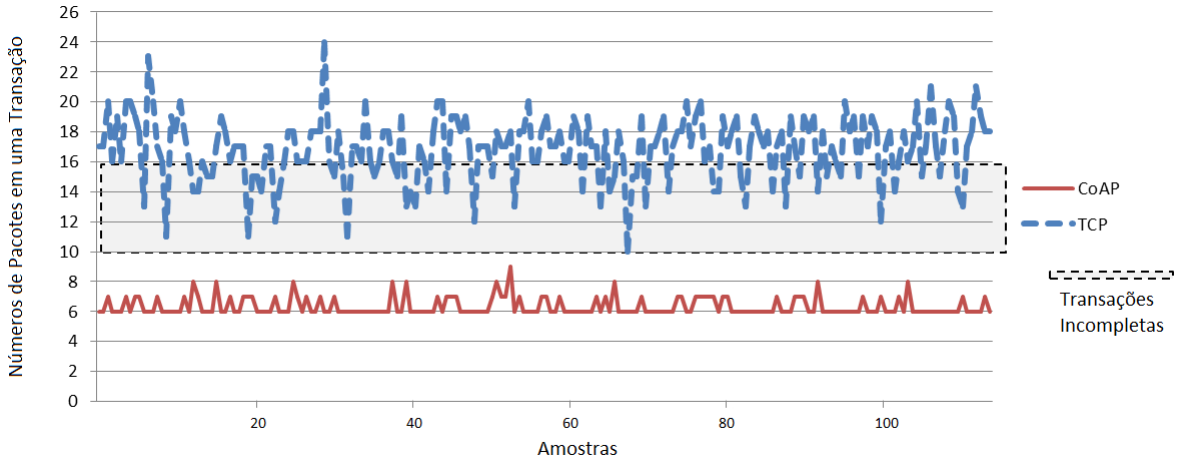


Figura 4.7: Resultados da transmissão em um canal com 10% de perda de pacotes.

Esses resultados mostram que, mesmo com pacotes perdidos, a solução CoAP troca menos pacotes que a solução TCP/IP. Outro ponto interessante a observar é que todas as transações ISO/IEEE 11073 CoAP foram completadas, ao contrário da solução TCP/IP. Ao utilizar sockets TCP/IP algumas transações não são completadas, e precisam ser reiniciadas. Esse comportamento ocorre dada consecutivas perdas de pacotes durante uma mesma transação, o que força o socket TCP/IP a desconectar e, consequentemente, faz com que toda a transação ISO/IEEE 11073 seja abortada. Uma das razões para que o socket seja desconectado se dá pelo fato que uma das pontas da conexão considera o canal como não disponível e, portanto, fecha a conexão socket.

O último teste realizado teve como objetivo avaliar a utilização do *11073Server* e *11073Client* sobre diferente interface de comunicação sem fio. Para esses testes o *11073Server* foi instalado em uma máquina em uma infraestrutura em nuvem disponibilizada pela Amazon[5], e três interfaces de comunicação sem fio foram utilizadas no *11073Client*:

- uma interface Wi-Fi IEEE 802.11g com uma ponto de acesso conectado a uma conexão

[5]http://aws.amazon.com

com a Internet aDSL de 30Mbps;

- uma interface de comunicação celular com conectividade 3G (HSDPA+);
- uma interface de comunicação celular com conectividade 2.5G (EDGE).

Um objetivo secundário desse teste foi avaliar como redes heterogêneas com a presença de *firewalls* e translação de redes (NAT) podem interferir em uma comunicação CoAP/UDP. A Figura 4.8 apresenta os resultados em termos de duração média de uma transação e sua variação.

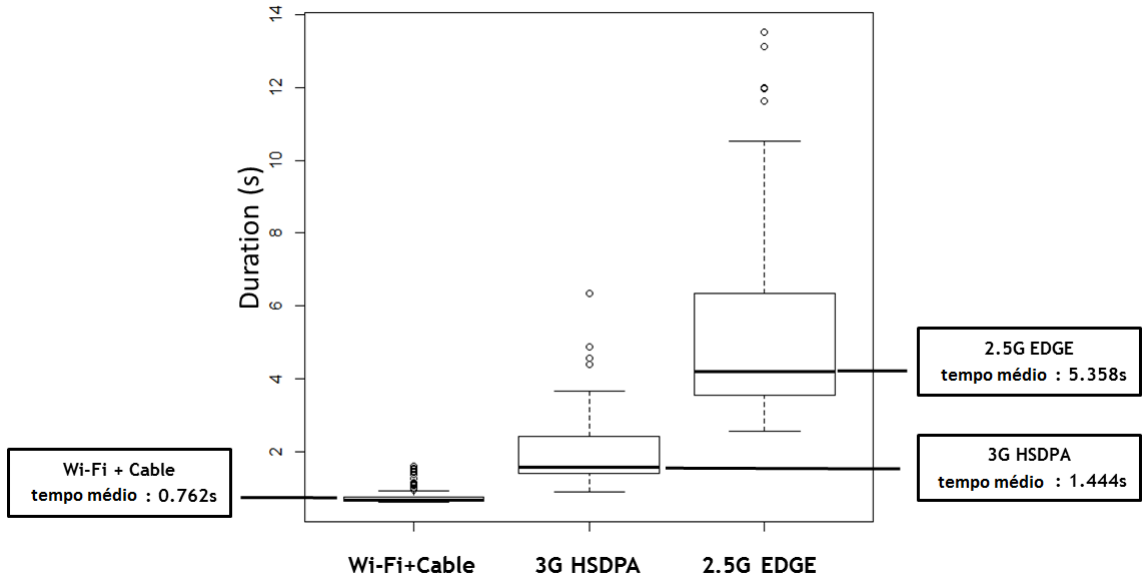


Figura 4.8: Transações CoAP IEEE 11073 em diferentes meios transporte.

É importante ressaltar que todas as transações ISO/IEEE 11073 foram completadas e, portanto, a transmissão através de redes heterógenas com *firewalls* não apresentou limitações para a solução proposta. Em relação aos resultados, é possível observar que as conexões Wi-Fi/aDSL e 3G apresentaram uma média capaz de suprir os requisitos de QoS para aplicações de monitoramento (atraso menor que 3 segundos), os quais são definidos no documento ISO/IEEE 11073-00101 [72].

### 4.1.5 Integração e Testes de Interoperabilidade

Como mencionado anteriormente, a solução CoAP foi integrada com outras soluções de saúde conectada para avaliação. O primeiro sistema a ser integrado e avaliado foi o *UPnP*

*Health*. Nesse sistema foi desenvolvido um UPnP *HealthGateway* utilizando um agregador ISO/IEEE 11073 disponibilizado pela biblioteca Antidote, chamado de *health-d*. Esse agregador coleta dados de DPS utilizando a interface Bluetooth HDP do Linux, e compartilha dados pré-processados através de uma interface padrão no sistema operacional. Esses dados são codificados utilizando um formato provido pela biblioteca Antidote, chamado de *DataList*. Na camada de comunicação UPnP foi utilizado o arcabouço de desenvolvimento UPnP Brisa, o qual recebe e compartilha dados utilizando serviços UPnP. Mais detalhes sobre o *UPnP Health* são apresentados em [35].

Para os testes de integração, foi desenvolvido um agregador CoAP IEEE ISO/11073 no mesmo dispositivo do UPnP *HealthGateway*. O objetivo foi possibilitar que DPS CoAP compartilhem dados com dispositivos em redes locais que tenham conectividade UPnP, como *Smart-TVs* e videogames. Foi utilizado o DPS oxímetro com conectividade CoAP ISO/IEEE 11073 apresentado na Figura 4.5. Além do DPS CoAP, dispositivos Bluetooth foram utilizados para comunicação direta com o UPnP *HealthGateway* através do serviço UPnP *LocalHealth*. Esses dispositivos foram utilizados para avaliar como um agregador ISO/IEEE 11073 único se comporta com múltiplas requisições de diferentes dispositivos em diferentes interfaces de comunicação. A Figura 4.9 apresenta o ambiente de avaliação utilizado, e alguns dos dispositivos utilizados. Para cada DPS, um conjunto de dez (10) operações foi realizado. Como resultado geral, todas as operações foram realizadas com sucesso, e a integração e comunicação entre os dispositivos na rede UPnP e CoAP ocorreu com sucesso, de modo que dispositivos CoAP foram listados por *Control Points* UPnP.

O outro teste realizado foi a integração de um CoAP *11073Server* com um sistema de Saúde Conectada comercial, chamado de SigHealth[6]. Esse sistema é compatível com as recomendações do *Continua Health Alliance*, e sua arquitetura pode ser ilustrada na Figura 4.10. O principal propósito para a utilização do SigHealth foi avaliar o uso da nova arquitetura em conjunto com uma plataforma já disponível no mercado. Nessa plataforma, por exemplo, agregadores estão disponíveis para diferentes plataformas móveis, como o Android do Google, o iOS da Apple e o Linux. Além disso, a interface PAN suporta diversas tecnologias, como o Bluetooth HDP, USB, ou o Bluetooth Low-Energy através da transcodificação de dados ISO/IEEE 11073 [74].

[6]http://health.signove.com

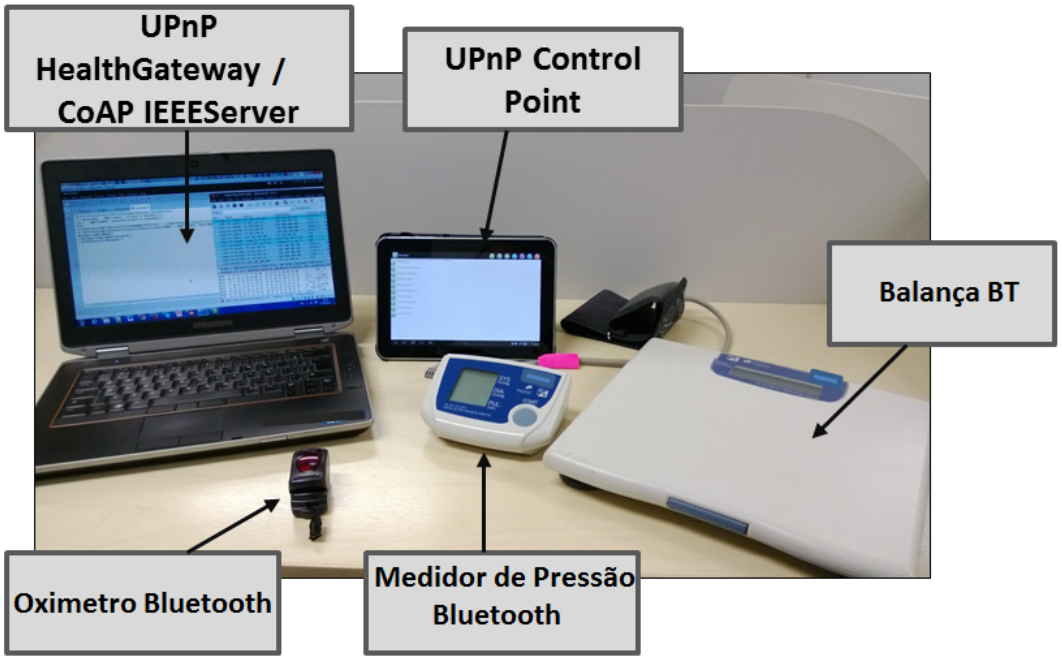


Figura 4.9: Dispositivos e ambiente de avaliação UPnP.

Na integração do CoAP *11073Server* foi desenvolvido um Agregador de Saúde na Internet (ASI), o qual foi integrado através de serviços Web disponibilizados pelo SigHealth, como ilustrado na Figura 4.11.

Os testes realizados nessa configuração tiveram como objetivo avaliar como um serviço já disponibilizado na Internet se comporta com a nova solução proposta e o protocolo CoAP. Foram criadas múltiplas instâncias de um ASI integrado aos serviços Web do SigHealth. Cada instância do ASI faz a translação entre mensagens ISO/IEEE 11073 e o formato de dados do SigHealth. Como as interfaces do SigHealth seguem as recomendações do *Continua Health Alliance*, suas mensagens são baseadas no padrão HL7 [75]. A Figura 4.12 apresenta o fluxo de dados entre um ASI e os serviços do SigHealth. Para cada ASI, testes foram executados onde um mesmo usuário compartilha dados utilizando dispositivos CoAP ISO/IEEE 11073 e DPS legados já suportados pela plataforma SigHealth. Em todos os testes, todos os dados foram recebidos e consolidados no sistema de armazenamento de dados de saúde do SigHealth.

Alguns casos de usos foram definidos para avaliar a relevância da arquitetura proposta. Os mais interessantes são apresentados e discutidos nas próximas seções.

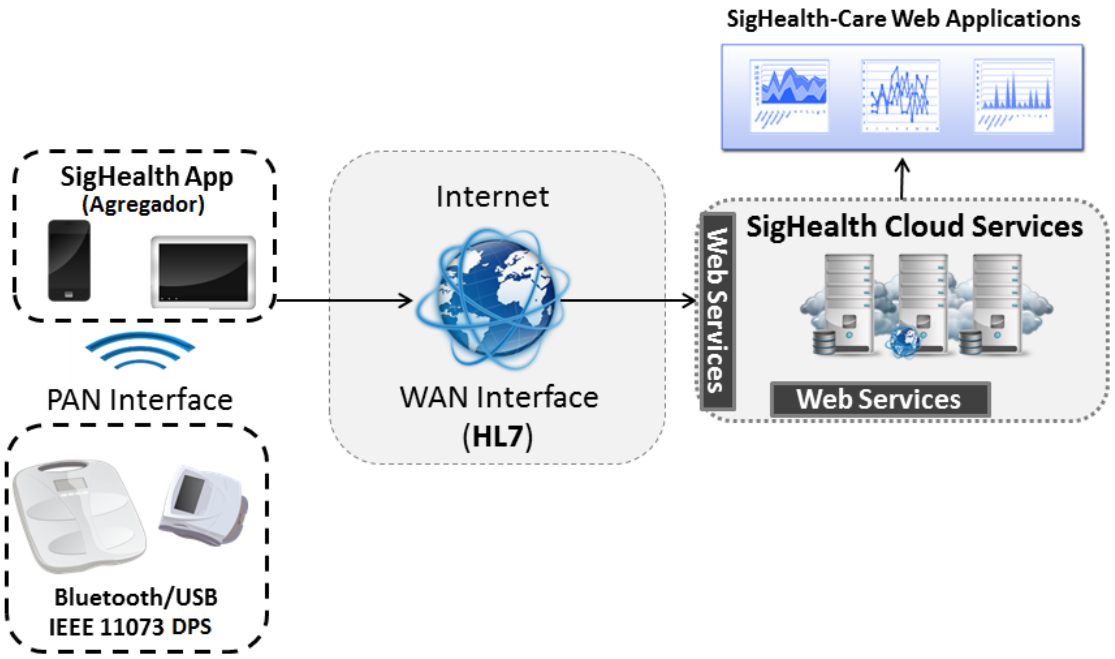


Figura 4.10: Diagrama com a arquitetura de um Sistema de Monitoramento Remoto de Pacientes na Internet.

### Compartilhando Dados de Saúde Diretamente para Internet

A Figura 4.13 ilustra um caso onde um DPS conecta-se com um serviço na Internet utilizando um *smartphone*. É interessante ressaltar que o *smartphone* não precisa necessariamente ter instalado algum tipo de serviço ou aplicação para cuidados de saúde. Ao utilizar a arquitetura proposta, o *smartphone* apenas precisa ter instalado um serviço de Gateway de Internet em alguma de suas interfaces sem-fio, como o Bluetooth Low-Energy. Como uma extensão desse caso de uso, o mesmo DPS pode ser utilizado em outros ambientes da casa, e ao invés de se conectar com o *smartphone* o mesmo pode se conectar com uma *Smart-TV* que oferece o mesmo serviço de Gateway de Internet, como ilustrado na Figura 4.13(b).

Alguns questionamentos devem ser levados em consideração nesse caso de uso:

- como garantir que o Gateway de Internet é confiável?
- por que criar um servidor CoAP na nuvem?

Para a primeira questão, uma solução é autorizar cada Gateway individualmente, por exemplo, utilizando o mecanismo de pareamento do Bluetooth. Para a segunda questão, ao

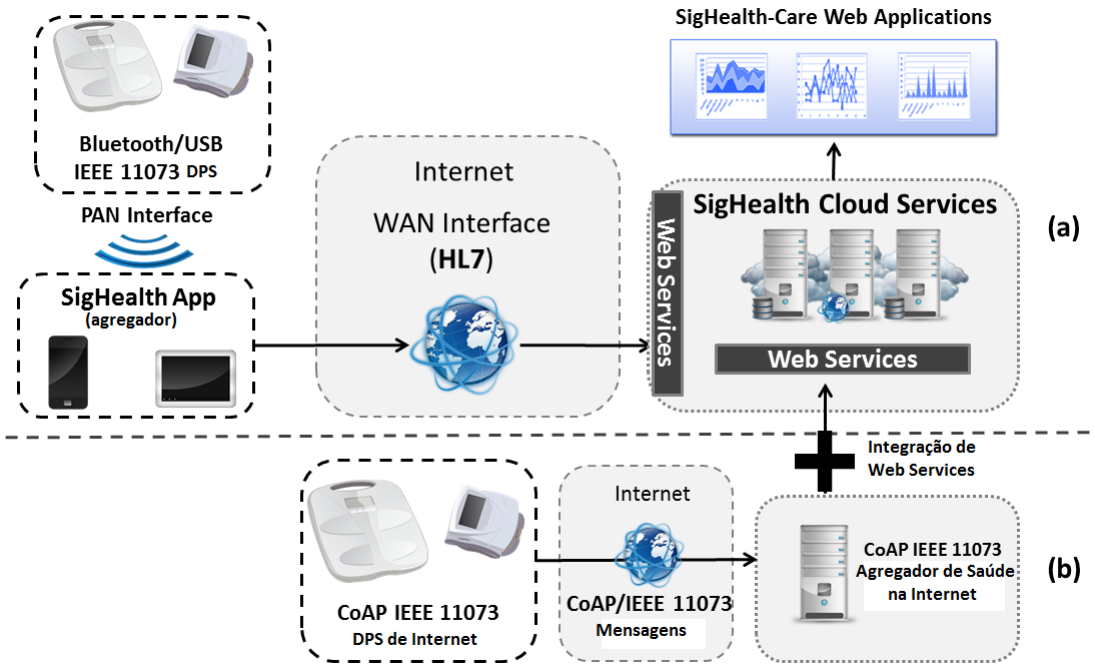


Figura 4.11: Diagrama com arquitetura estendida de um Sistema de Monitoramento Remoto de Pacientes na Internet no mesmo servidor.

invés de criar um servidor CoAP na nuvem, é possível fazer mapeamento de mensagens CoAP para mensagens HTTP no Gateway através de um proxy [43][33]. Dessa maneira, é possível utilizar o mesmo servidor para clientes CoAP e HTTP.

### Utilizando Sensores e Atuadores Corporais para Tratamentos de Saúde

Nesse cenário, um paciente utiliza sensores e atuadores corporais durante um tratamento de saúde. Por exemplo, o paciente pode ter um glicosímetro portátil sob a pele, um monitor cardíaco em seu pulso, e uma bomba de insulina portátil que é ativada por uma aplicação instalada em seu *smartphone*. A aplicação de saúde periodicamente checa os sinais vitais do paciente e decide qual o melhor momento para ativar a bomba de insulina. Nesse cenário é importante para a aplicação interpretar os dados de saúde o quanto antes, para que seja possível tomar a decisão no momento correto. Portanto, é importante que o Gateway no *smartphone* compartilhe os dados com as aplicações locais e com os serviços na Internet, como descrito no Capítulo 1.

Algumas limitações se aplicam para esse cenário. Um Gateway de Internet executando

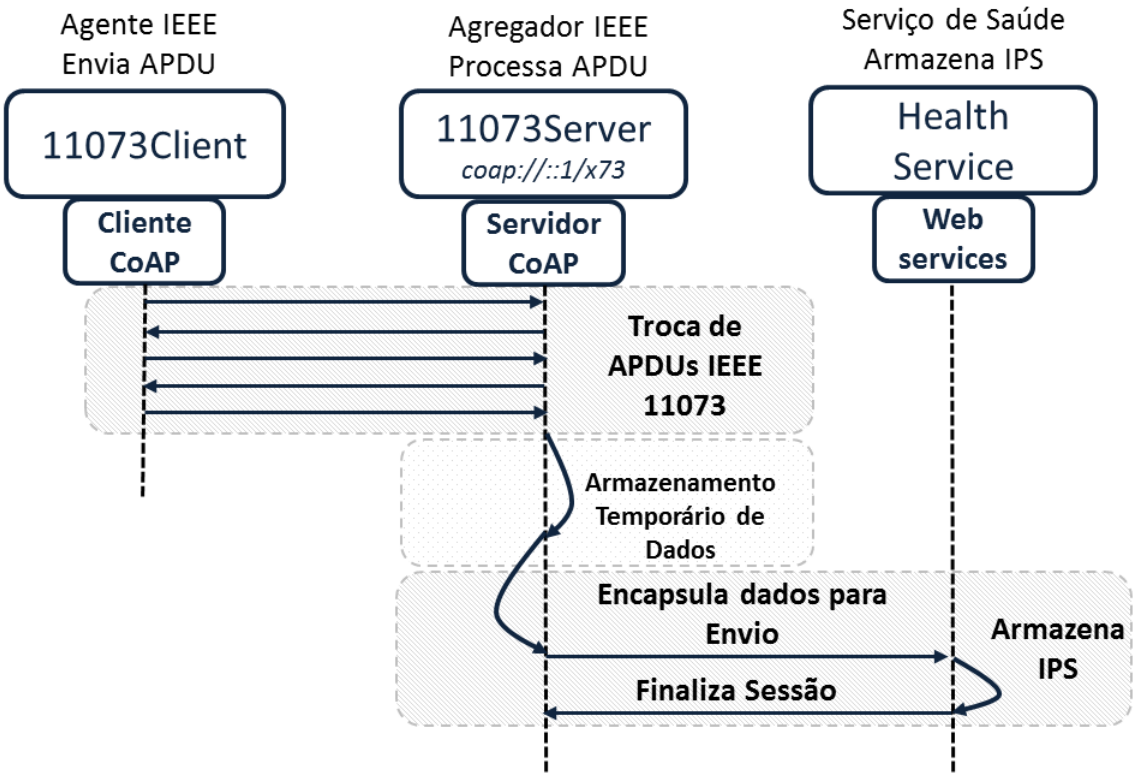


Figura 4.12: Diagrama com Procedimento de Integração com um Serviço de MRP na Internet.

no mesmo dispositivo de um agregador de saúde pode resultar em cenários onde a mesma interface de rede é compartilhada com diferentes serviços. Por exemplo, diferentes dispositivos podem estar utilizando o mesmo serviço de Gateway de Internet. Esse compartilhamento de interface de comunicação pode resultar em complicações em situações de emergência, como vai ser apresentado nas próximas seções.

## 4.2 Protótipo de um Gateway Bluetooth Low-Energy

A arquitetura apresentada na seção anterior descreve uma solução para comunicação entre DPS e serviços na Internet através do uso de um novo modelo de comunicação e o protocolo CoAP. Como também descrito anteriormente, o CoAP pode ser executado sobre diferentes meios de comunicação, como o protocolo UDP e até mesmo SMS. Considerando que DPS são dispositivos simples e com limitações de processamento, memória e consumo de energia, esses, em sua maioria, irão fazer uso de meios de comunicação sem fio de baixo consumo,

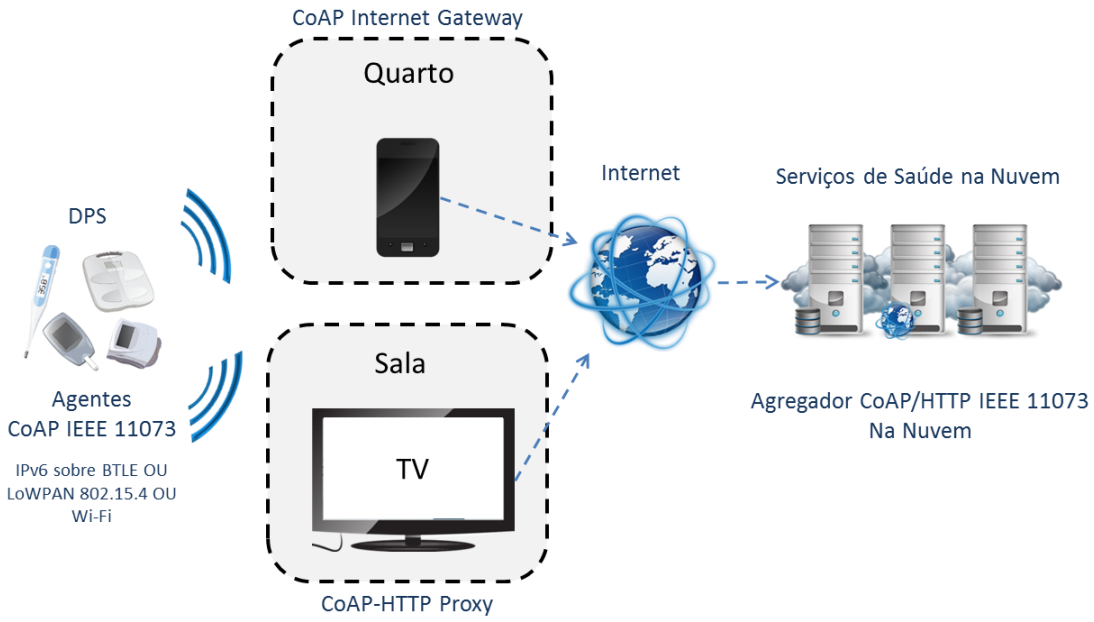


Figura 4.13: Diagrama com DPS CoAP utilizando Internet Gateways.

como ZigBee[7], ANT+ [8], e o Bluetooth Low-Energy (BLE) [9]. A maioria dessas tecnologias suportam especificações para a transmissão de dados IP sobre suas interfaces físicas. Ao utilizar essas tecnologias, considerando uma abordagem simples, faz-se necessária a utilização de Gateways para transportar os pacotes IP de um meio físico de transmissão para o outro. Como introduzido no Capítulo 1, um Gateway instalado em um dispositivo pessoal é responsável pela coleta e encaminhamento de dados de sensores para outra rede com conectividade com a Internet.

Portanto, a utilização de Gateways pessoais, onde o Gateway é utilizado apenas por sensores e dispositivos pertencentes a mesma pessoa, torna-se um caso de uso comum para o usuário final.

Nessa seção foi desenvolvido um protótipo de Gateway para a transmissão de pacotes IP utilizando o Bluetooth Low-Energy (BLE). O BLE foi escolhido por sua ampla adoção em diversos tipos de dispositivos de uso doméstico e pessoal, como *smartphones*, *tablets* e *Smart-TVs*, além de suas características de baixo consumo [14]. Para esse primeiro protótipo foi desenvolvido um perfil IP para utilização sobre o Bluetooth GATT [9]. Essa abordagem

[7]http://www.zigbee.org
[8]www.thisisant.com

foi escolhida dado que a especificação do perfil IP para o Bluetooth Low Energy foi lançada apenas em Dezembro de 2014. O modelo GATT adotado define duas característica GATT para pacotes IP no dispositivo cliente. Uma característica suporta escrita para o recebimento de pacotes IP, e a outra característica suporta leitura e notificações para o envio de pacotes IP. O dispositivo BLE servidor escreve e recebe notificações de pacotes IP dos DPS, e faz o roteamento interno para Internet, como apresentado no fluxo ilustrado na Figura 4.14.

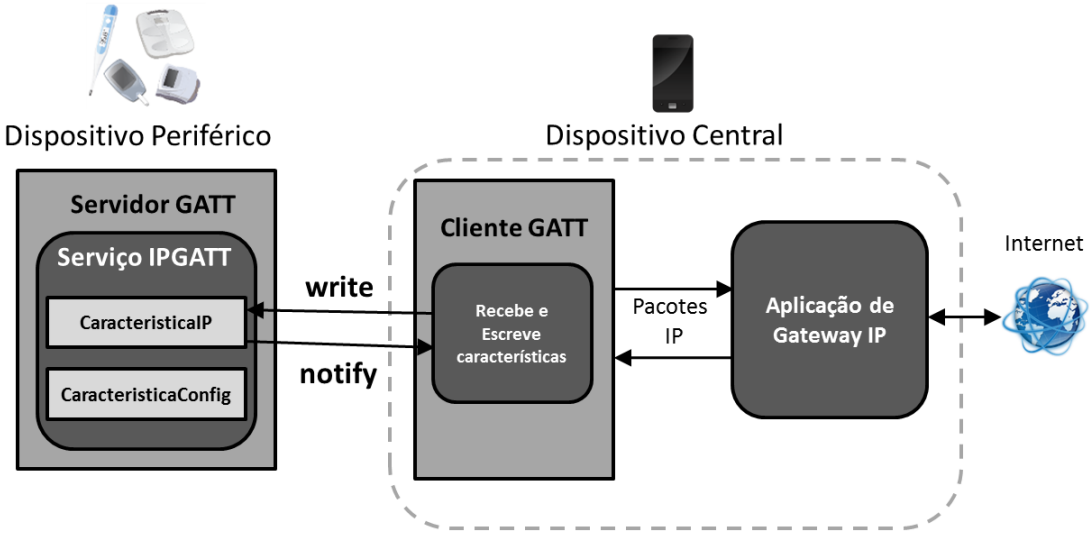


Figura 4.14: Diagrama com modelo de Características GATT para Roteamento de Pacotes IP.

Com esse protótipo BLE foi possível realizar testes de avaliação da arquitetura em redes pessoais de baixo consumo.

### 4.2.1 Avaliação Experimental em uma Rede BLE

Testes foram realizados com o objetivo de avaliar a rede BLE desenvolvida para o protótipo. Em especial, uma característica de redes Bluetooth e BLE motivou a realização desses testes. Em geral, essas redes são compartilhadas por vários dispositivos e serviços. Por exemplo, uma rede BLE com suporte a IP poderá se conectar com DPS e dispositivos de multimídia ao mesmo tempo. Esse compartilhamento, de maneira geral, faz com que a taxa de transmissão média de cada dispositivo diminua dado que todos estão acessando o mesmo meio, e o BLE irá compartilhar o meio de transmissão de maneira homogênea utilizando canais *best-effort*.

Entretanto, alguns DPS necessitam de uma taxa de transmissão mínima para o envio de dados, pois esses dados são coletados continuamente pelo módulo de sensoriamento, e o DPS tem uma memória limitada (buffer) para a contenção de dados até o seu envio. Caso a taxa de transmissão não seja contemplada, o buffer de contenção irá ser preenchido, e novos dados de saúde coletados serão perdidos. Por exemplo, considere um DPS e um canal com as seguintes características:

**Definição 1 (Modelo de Dados de um DPS)** .

$$\textit{Taxa de dados} = Tx_{min} \textit{ em bytes/segundos}$$
$$\textit{Buffer interno} = B_{in} \textit{ em bytes}$$

**Definição 2 (Canal Disponível)** .

$$\textit{Taxa de transmissão disponível} = Tx_{canal} \textit{ em bytes/segundos}$$
$$\textit{Onde: } Tx_{canal} < Tx_{min}$$

Portanto, o buffer interno $B_{in}$ irá completar e dados coletados serão perdidos após o seguinte período de tempo:

**Definição 3 (Limite do Buffer Interno)** .

$$\textit{Período até o Buffer Preencher: } P_{max} = B_{in}/(Tx_{min} - Tx_{canal}) \textit{ em segundos}$$

Ao completar o buffer interno com dados, o DPS tem as seguintes opções:

- diminuir a taxa de coleta de dados dos sensores. Entretanto, essa abordagem leva a uma menor precisão do conjunto de dados coletados. Por exemplo, uma curva de ECG com menos pontos disponíveis;
- descartar dados coletados dos sensores. Novos dados coletados pelo sensor serão descartados até que exista espaço no buffer para transmissão;
- desconectar, e esperar que o canal disponibilize a banda de transmissão necessária.

Em todos os casos, o DPS vai deixar de enviar dados de sinais vitais aos serviços e aplicações de saúde. Um experimento foi realizado para demonstrar esse problema. Um simulador de ECG foi criado, o qual envia dados a $Tx_{min} = 16.5Kbytes/s$ em uma rede Bluetooth. Essa mesma rede tem uma capacidade máxima de $C_{max} = 50Kbytes/s$ a ser compartilhada com todos os dispositivos conectados. Foram criados fluxos paralelos que consomem uma banda constante de $Tx_{outros} = 35Kbytes/s$. Para esses fluxos, dois dispositivos genéricos foram criados, os quais se conectam com o Gateway BLE, e enviam dados de maneira serial ao mesmo. Portanto, a banda de transmissão disponível para um terceiro dispositivo, o simulador de ECG, é $Tx_{max} = C_{max} - Tx_{outros} = 15Kbytes/s$.

O buffer interno do simulador de ECG é o existente no dispositivo de hardware Bluetooth, no caso específico desse experimento, um dispositivo Bluetooth USB com buffer de $B_{in} = 8Kbytes$.

O simulador ECG gera dados e pacotes CoAP ISO/IEEE 11073 e os envia a interface Bluetooth constantemente . Quando o $B_{in}$ é completado, a interface Bluetooth desconecta automaticamente para evitar que ocorra um *buffer overflow* em camadas inferiores. O gráfico da Figura 4.15 apresenta esse comportamento, onde após um período de $P_{max}$ $5,3s$ do ponto (a) ao ponto (b), o dispositivo ECG desconecta por não conseguir dar vazão aos dados gerados pelo simulador.

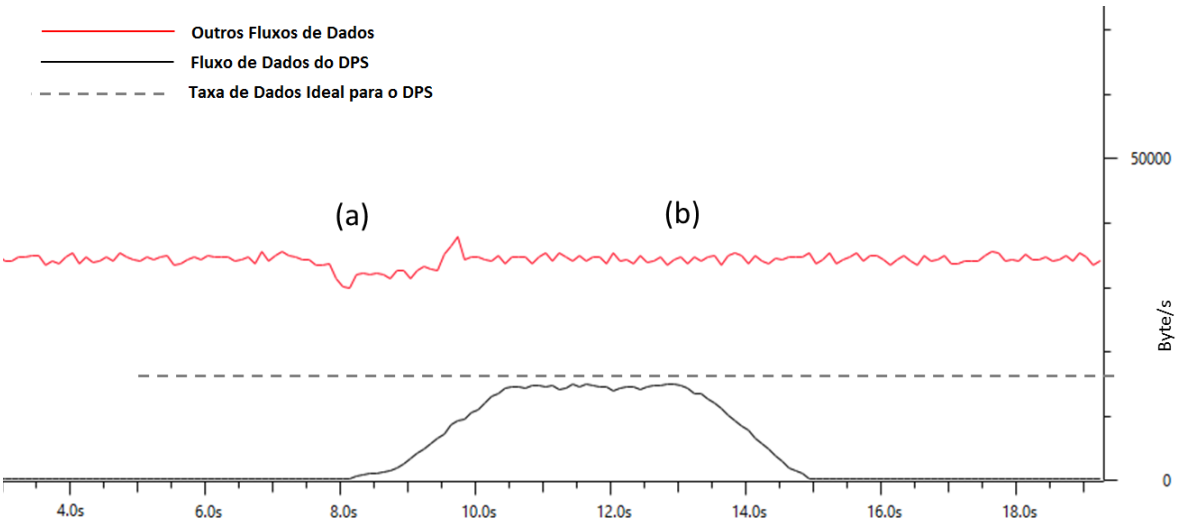


Figura 4.15: Fluxos de dados em um Gateway Bluetooth comum.

Tratando-se de aplicações e serviços para saúde do paciente, esse tipo de comportamento

não é aceitável em algumas situações. Por exemplo, quando o paciente estiver passando por um procedimento de monitoramento onde é necessário ter acesso a dados em tempo real, dados originários dos DPS que estão realizando esse procedimento devem ter prioridade em relação aos outros.

## 4.3 Monitor de parâmetros de QoS para Fluxos de Saúde

Com o problema apresentado anteriormente, nesse trabalho é proposto utilizar um Monitor de parâmetros de QoS para Fluxos de Saúde (MQS), criando um *Smart-Gateway* para Saúde. O principal objetivo do MQS é analisar o tráfego de rede de maneira passiva, e identificar situações onde o fluxo de dados de um DPS requer priorização ou não. Com isso, o MQS também é responsável por acessar a camada de controle da interface de comunicação na rede PAN, e desconectar serviços e dispositivos que não são de interesse ao procedimento de monitoramento de saúde em execução naquele momento. A Figura 4.16 apresenta os principais módulos existentes no MQS.

Três módulos principais compõem o núcleo do MQS:

- *Registro de Dispositivos (RD)*. Esse módulo tem o registro de informações relativas a cada DPS registrado no *Smart-Gateway* para Saúde. Essas informações incluem a taxa mínima requerida durante um fluxo de dados, o tamanho do seu buffer interno, e o tipo de DPS;

- *Interpretador de Regras de Monitoramento (IRM)*. Esse módulo contém regras que definem quando um determinado *Smart-Gateway* está recebendo fluxo de dados referente a um monitoramento. Por exemplo, uma regra pode definir que toda vez que um fluxo de ECG for iniciado, significa que o *Smart-Gateway* está participando de um processo de monitoramento, e o fluxo ECG deve ser priorizado;

- *Avaliador de Fluxos de Saúde (AFS)*. Esse módulo é responsável por receber o fluxo de dados enviado por um monitor de tráfego no *Smart-Gateway* para Saúde (2), e extrair informações do ISO/IEEE 11073, funcionando como um agregador de saúde passivo. Esse módulo atua de maneira passiva, pois não troca informações com o agente, portanto, ele utiliza as informações trocadas pelo agente e outro agregador para interpretar

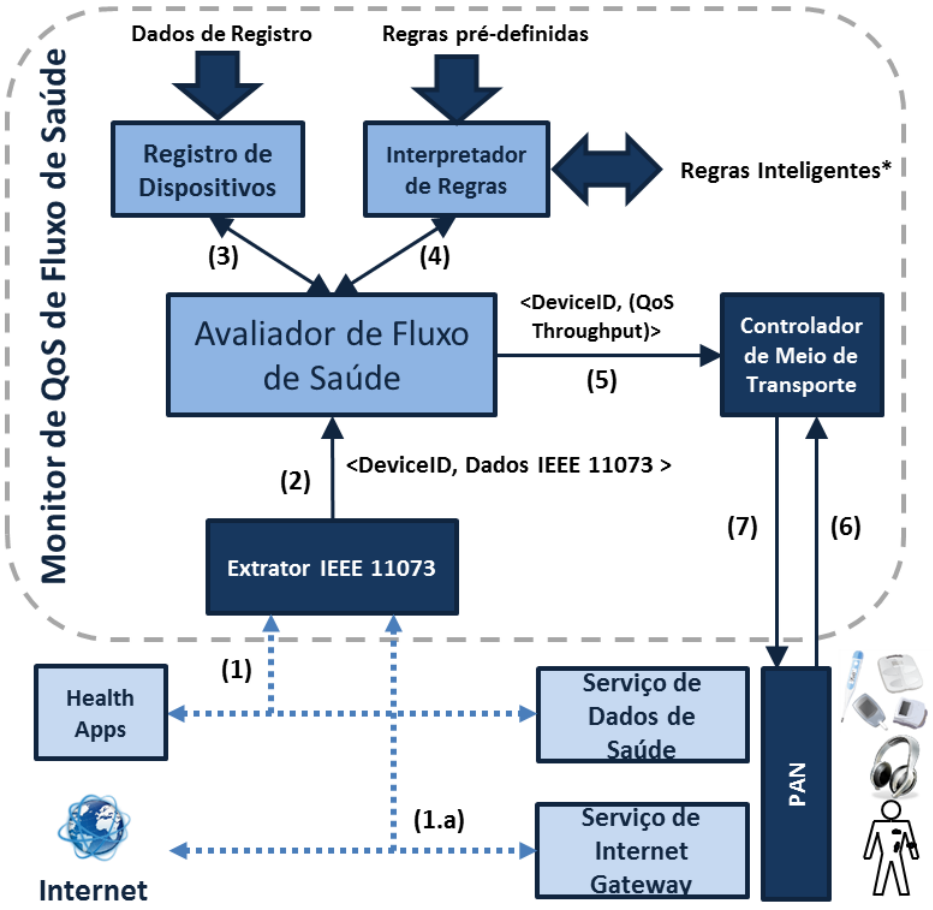


Figura 4.16: Diagrama com visão geral de um módulo MQS em um Smart-Gateway.

os dados ISO/IEEE 11073. Por fim, esse módulo faz uso das informações do RD (3) e IRM (4) para definir os requisitos de rede mínimos para cada DPS participante de um monitoramento.

Outros dois módulos secundários e dependentes da plataforma são necessários para completar o MQS:

- *Extrator de Pacotes ISO/IEEE 11073 (11073Ex).* Esse módulo é integrado diretamente ao *Smart-Gateway* para Saúde, e extrai APDUs ISO/IEEE 11073 dos pacotes CoAP para processamento pelo MQS (1). Apesar de ser integrado ao *Smart-Gateway* para Saúde, esse módulo pode ser inserido em outros Serviços de Gerenciamento de Dispositivos de Saúde na plataforma alvo, como um agregador Bluetooth HDP;

- Controlador do Meio de Transporte (CMT). Esse controlador tem privilégios para controlar a interface de comunicação sem fio da plataforma. O CMT, portanto, recebe comandos do AFS (5) e, dado a configuração da rede e uma estimativa da banda disponível, realiza operações para liberar banda para os dispositivos descritos no comando do AFS (6 e 7). Por exemplo, o AFS pode enviar um comando para dar prioridade ao dispositivo $Device_A$ que necessita de uma banda $Tx_A$. O CMT verifica se a rede tem banda disponível para esse dispositivo, caso negativo, o CMT toma a decisão de desconectar outros dispositivos que não fazem parte do comando enviado pelo AFS.

### 4.3.1 Avaliação de um Protótipo de MQS em uma Rede BLE

Para a avaliação do MQS foi utilizada a mesma configuração de rede e dispositivos do experimento anterior. No desenvolvimento do MQS para avaliação, os módulos IRM e RD têm valores e regras pré-fixados. No RD o dispositivo ECG $Device_{ECG}$ foi configurado com seus valores de $B_{in} = 8Kbytes$ e $Tx_{min} = 16.5Kbytes/s$. O módulo IRM teve apenas uma regra registrada indicando que quando um fluxo ECG inicia, significa que o *Smart-Gateway* está em monitoramento. Para o registro de regras foi utilizada a nomenclatura ISO/IEEE 11073:10101 [13] para identificação de fluxos e dados de saúde, e uma linguagem simples para composição de regras. Por exemplo, a regra a seguir descreve o caso anterior:

Código Fonte 4.1: Exemplo de uma Regra Simples de Identificação de Monitoramento

```
rule01:
when {MDC_DEV_SUB_SPEC_PROFILE_ECG starts}
then {monitoring starts}
```

Por fim, o módulo CMT recebe instruções do AFS para priorizar o $Device_{ECG}$ com uma taxa $Tx_{min} = 16.5Kbytes/s$, e que esse dispositivo tem um $B_{in} = 8Kbytes$. Com essa informações, o CMT faz uma estimativa do tempo $P_{max}$ necessário para o buffer ser preenchido seguindo a Definição 3. Com esse $P_{max}$ estimado, o CMT define um período de espera $P_{limit}$ de até 80% de $P_{max}$ para que a rede disponibilize $Tx_{min}$ para $Device_{ECG}$ . Isso pode ocorrer caso algum outro dispositivo voluntariamente desconecte e libere banda. Caso $P_{limit}$ se esgote e $Tx_{min}$ não esteja disponível para $Device_{ECG}$, o módulo CMT atua na rede para desconectar outros dispositivos a fim de liberar banda para os dispositivos participantes do

processo de monitoramento. O módulo CMT protótipo foi desenvolvido com uma abordagem simples, onde todos os fluxos que não fazem parte das instruções de monitoramento do MQS são desconectados. O gráfico da Figura 4.17 apresenta o comportamento dos fluxos de dados com o uso do MQS.

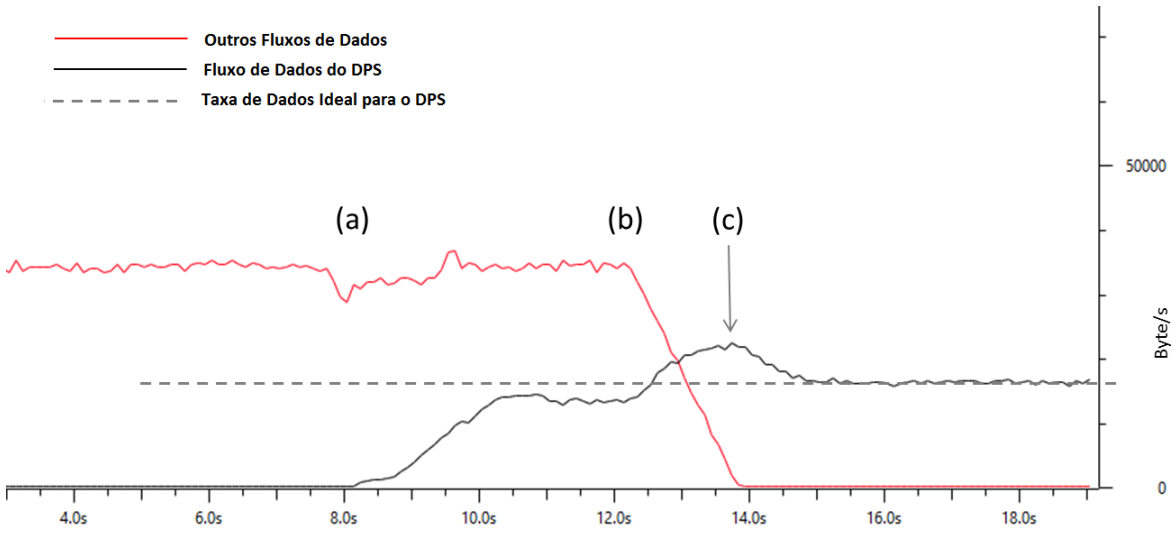


Figura 4.17: Fluxos de dados utilizando um protótipo de MQS.

No gráfico da Figura 4.17 alguns pontos merecem destaque. No ponto (a) o $Device_{ECG}$ inicia seu fluxo de dados. Nesse mesmo instante, o MQS identifica que é um fluxo de monitoramento através da avaliação da regra em IRM e envia instruções para o CMT. O CMT calcula um $P_{limit}$ $4,24s$, e no ponto (b) desconecta todos os outros dispositivos presentes na rede. Por fim, no ponto (c) é observado um fluxo maior de dados originado do $Device_{ECG}$ devido ao envio dos pacotes que estavam armazenados no buffer interno do dispositivo.

## 4.4 Considerações Finais do Capítulo

Neste capítulo foi apresentada uma arquitetura para um sistema de MRP, o qual serve de base para o trabalho apresentado nesse documento. Na definição dessa arquitetura, alguns requisitos como interoperabilidade e possibilidade de exportar dados para a Internet foram considerados. Com os resultados experimentais obtidos, o sistema apresentado se apresenta como uma solução viável para o compartilhamento de IPS na Internet de maneira padronizada.

Além dos resultados experimentais obtidos, a partir da definição e desenvolvimento da arquitetura proposta, requisitos detalhados para a validação do problema de controle de fluxo de dados de saúde foram obtidos. Com resultados experimentais foi demonstrado que é necessário disponibilizar recursos mínimos de QoS para DPS participantes de um processo de MRP, de modo a evitar que os mesmos abortem o processo, por exemplo.

Um protótipo de um *Smart-Gateway* foi implementado nessa primeira versão da arquitetura, de modo a validar que o processo de avaliação de contexto pode contribuir em um processo de controle de fluxo.

# Capítulo 5

# Controle de Fluxo Adaptativo para Gateways Bluetooth Low-Energy

Como introduzido no Capítulo 4, a utilização de Gateways de Internet em Sistemas de Monitoramento Remoto de Pacientes podem criar desafios relativos a limitação da tecnologia aplicada nesses Gateways. Considerando a tecnologia Bluetooth Low-Energy (BLE) como tecnologia alvo nesse trabalho, e realizando um estudo sobre a transmissão de dados IPv6 sobre BLE, foram observadas limitações tecnológicas que devem ser consideradas durante o projeto de Gateways de Internet para o BLE.

Neste capítulo, portanto, é apresentado o projeto de um *Smart-Gateway* Bluetooth Low-Energy. Inicialmente é realizado um estudo sobre como a limitação tecnológica de um controlador de hardware BLE pode afetar aplicações e dispositivos que fazem uso de Gateways de Internet com BLE. Em seguida é detalhado o processo de desenvolvimento de um controlador para Gateways IPv6 BLE, o qual leva em consideração requisitos de QoS dos dispositivos, e características de prioridade temporal controladas por aplicações e serviços no próprio *Smart-Gateway*. Com isso, o *Smart-Gateway* apresentado neste capítulo tem como objetivo ser aplicado ao Sistema de Monitoramento Remoto de Pacientes apresentado no Capítulo 4.

## 5.1 Visão Geral do Problema e Motivação

Como apresentado no Capítulo 2, um dispositivo Bluetooth pode ser dividido em duas grandes camadas, o hospedeiro (*host*) e o controlador (*controller*). Essas duas camadas são divi-

didas por uma interface chamada *Host-Controller Interface* (HCI). De maneira generalista, o controlador pode ser considerado como o hardware Bluetooth, pois nessa camada é onde se encontra todo o controle da camada física, incluindo o controlador do rádio Bluetooth. Do lado oposto ao controlador do rádio Bluetooth, existe algum tipo de interface de entrada e saída (E/S) de dados, o qual pode ser uma interface USB ou serial por exemplo.

Por ser um dispositivo simples, o controlador Bluetooth impõe restrições, as quais podem ser relativas a decisões de custo de desenvolvimento e produção, além de outros fatores como, por exemplo, o baixo consumo de energia. Portanto, a evolução dinâmica de um controlador Bluetooth torna-se mais complexa em relação ao lado hospedeiro da pilha de protocolos Bluetooth. Por exemplo, pode-se considerar mais simples atualizar a camada L2CAP do Bluetooth em uma distribuição Linux, do que ter que atualizar o firmware de um controlador Bluetooth com interface USB.

Nesse sentido, dada a característica dinâmica que novas aplicações e serviços impõem ao controlador Bluetooth, mecanismos de controle devem ser criados e adaptados nas camadas superiores do lado hospedeiro na pilha de protocolos.

A Figura 5.1 apresenta um diagrama de um modelo simplificado de um controlador Bluetooth. Nessa figura pode-se observar que existem dois módulos de E/S de dados: o módulo de Rádio Bluetooth e o Módulo de Entrada e Saída do Controlador. Esses módulos impõem complexidades no gerenciamento de dados no Módulo Central do controlador. Esse Módulo Central deve ser capaz de fazer o gerenciamento de dados entres os dois módulos adjacentes fazendo uso de sua unidade controladora (MCU) e sua unidade de memória para o gerenciamento de *buffers*.

Além da limitação da banda de transmissão imposta por parâmetros temporais aplicados ao sistema de controle de acesso do Módulo de Rádio do Bluetooth, como apresentado nos trabalhos de [53] e [14], outras limitações são criadas por restrições no projeto de hardware do controlador Bluetooth. Por exemplo, o Módulo Central apresentado no diagrama da Figura 5.1 deve ser capaz de gerenciar o fluxo de dados vindouro do Módulo de Rádio e seus clientes e, além disso, gerenciar a demanda de dados requisitada pelo hospedeiro e suas aplicações e serviços.

Apesar dessa limitação de hardware do controlador Bluetooth ser visível, os projetos desses controladores de hardware Bluetooth, aparentemente, não a levam em consideração.

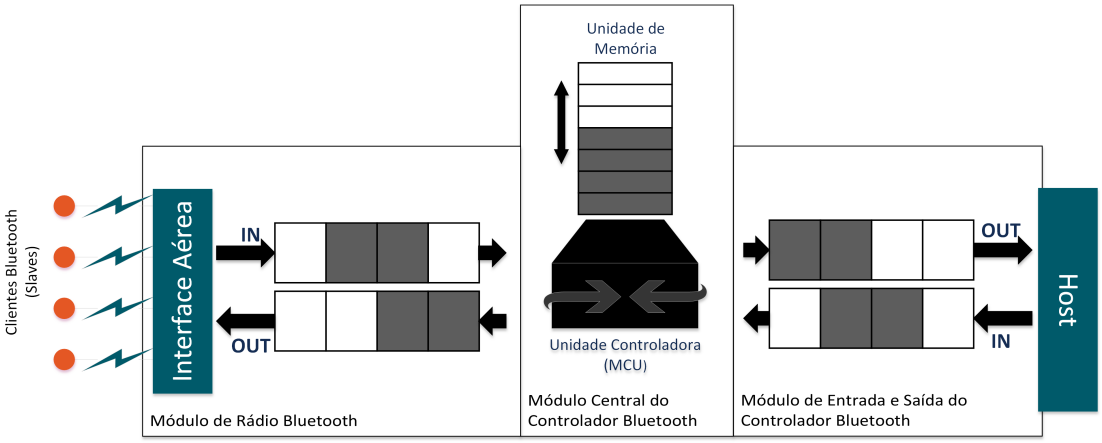


Figura 5.1: Diagrama de um modelo para um controlador Bluetooth.

Com isso, quando um controlador Bluetooth é exposto a uma situação de estresse, o mesmo pode entrar em um estado de instabilidade, levando a um consumo elevado no armazenamento de memória, ou a uma incapacidade de processamento de dados em tempo adequado em seu MCU. Na próxima seção são apresentados experimentos realizados com o propósito de expor essa limitação do controlador Bluetooth em diferentes configurações.

### 5.1.1 Avaliação do Problema

Para evidenciar a limitação de um controlador Bluetooth Low-Energy foram realizados experimentos com o propósito de explorar os limites de transmissão de dados em um enlace Bluetooth. Duas situações foram exploradas durante os experimentos:

- Definição do limite máximo da taxa de transmissão de dados em uma comunicação entre dois dispositivos Bluetooth Low-Energy.
- Observação do comportamento de uma *piconet* Bluetooth Low-Energy quando o dispositivo mestre atinge seu limite de transmissão de dados.

O ambiente de teste configurado durante os experimentos utilizou três diferentes modelos de Controlador Bluetooth como dispositivo mestre:

- Controlador Intel Wireless Bluetooth 7265 (Intel).
- Controlador Cambridge Silicon CSR8510 A10 (CSR).

- Controlador Broadcom BCM20702 (Broadcom).

Os dispositivos clientes sempre utilizaram o controlador CSR. O CSR foi escolhido devido a sua maior disponibilidade no mercado para compra, e também por este oferecer a maior taxa de transmissão fim-a-fim entre os modelos avaliados.

Como hospedeiros dos controladores Bluetooth foram utilizados computadores pessoais com o sistema operacional Linux. Desde a versão 3.19.0 o Kernel Linux oferece suporte a comunicação IPv6 sobre canais L2CAP do Bluetooth Low-Energy. Avaliações de diferentes versões do Kernel do Linux foram realizadas para a escolha da melhor configuração. Para a realização dos experimentos foi escolhida a versão 4.2.0-17 do Kernel como alvo para o dispositivo mestre devido a sua estabilidade na criação de redes *piconet*.

O primeiro conjunto de experimentos realizado teve como objetivo avaliar a taxa de transmissão máxima fim-a-fim que um dispositivo mestre suporta. Para a realização desse experimento foi estabelecido um canal de comunicação com controle de fluxo baseado em créditos para a transmissão de pacotes IPv6 sobre Bluetooth Low-Energy. Após a criação desse canal, o dispositivo cliente foi configurado para enviar pacotes IPv6 de maneira continua, aumentando sua taxa de dados gradualmente. Para a transmissão de dados IPv6, o dispositivo cliente fez o uso do comando *ping6*, de modo que o mesmo dado enviado no canal de *uplink* fosse retornado pelo canal de *downlink*.

O comando *ping6* também foi escolhido devido a seus parâmetros de configuração, o que permitiu configurar o tamanho de pacote e a espaço de tempo entre envios de dados. A configuração do tamanho do pacote serve para maximizar o uso dos créditos em relação ao pacote transmitido e o *Maximum Transmission Unit* (MTU) do canal. Ou seja, cada pacote transmitido pelo dispositivo cliente fez uso de um crédito e utilizou o tamanho de pacote máximo permitido. O ajuste do período de tempo entre o envio de pacotes serve para o ajuste gradual da taxa de transmissão do cliente.

O gráfico da Figura 5.2 apresenta o resultado de um experimento utilizando o Controlador Bluetooth Broadcom como dispositivo mestre. Neste gráfico são apresentados passos numerados indicando o crescimento da taxa de transmissão entre o cliente e o dispositivo mestre. No passo $4$ foi realizado um crescimento da taxa de dados, entretanto, o dispositivo mestre não foi capaz de processar esse aumento, e após alguns segundos a transmissão foi encerrada por parte do dispositivo mestre, sendo esse encerramento evidenciado pela queda

da taxa de transmissão ao final do gráfico. Esse experimento foi repetido para os três modelos de dispositivos como Controlador da rede Bluetooth Low-Energy.

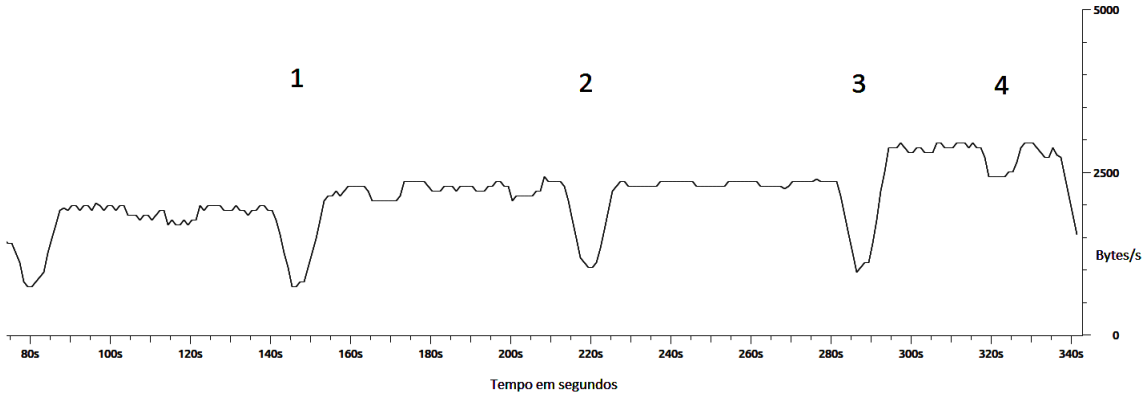


Figura 5.2: Resultados da taxa de transmissão máxima experimental em um nó mestre com chipset Broadcom BCM20702.

Para avaliar o fluxo de dados máximo que um dispositivo mestre suporta em nível de aplicação, ou seja, em nível IPv6, é necessário minimizar o impacto da troca de créditos (sinalização) entre o dispositivo mestre e os clientes. A configuração padrão do módulo Bluetooth do Kernel realiza uma troca de créditos entre dispositivos a cada 20 créditos consumidos por um cliente. Ou seja, a cada 20 pacotes enviados por um cliente, o dispositivo mestre utiliza o mesmo canal de transmissão para enviar um pacote de sinalização com a atribuição de novos créditos àquele cliente.

A Tabela 5.1 apresenta os valores máximos da taxa de transmissão suportada por modelo de Controlador Bluetooth como dispositivo mestre e com o valor padrão de 20 na renovação de créditos.

Tabela 5.1: Taxa de transmissão máxima suportada por modelo de Controlador Bluetooth.

| Modelo | Taxa de Transmissão Máxima |
|---|---|
| CSR8510 A10 | 10,8 Kbytes/s (86,4 Kbits/s) |
| Intel Wireless Bluetooth 7265 | 10 Kbytes/s (80 Kbits/s) |
| Broadcom BCM20702 | 2,6 Kbytes/s (20,8 Kbits/s) |

Além da avaliação do canal com o valor padrão de 20 para a renovação de créditos, foram realizados experimentos onde esse valor padrão foi alterado. O objetivo desses experimentos

foi a avaliar o impacto da troca de créditos durante uma transmissão fim-a-fim utilizando IPv6 sobre Bluetooth Low-Energy. Os resultados desses experimentos são apresentados no gráfico da Figura 5.3. Esse gráfico apresenta uma representação da interpolação dos resultados da taxa de transmissão máxima para diferentes valores de créditos para os três modelos avaliados.

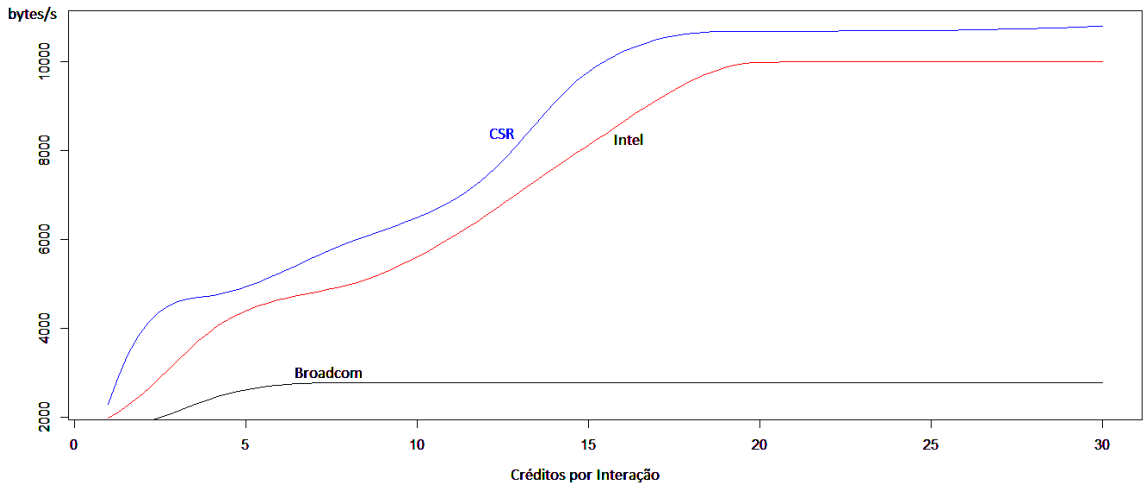


Figura 5.3: Variação da taxa de transmissão máxima em relação ao número de créditos trocados por interação.

É interessante observar o impacto na taxa de transmissão em nível de aplicação da troca constante de créditos durante uma transmissão. Por exemplo, ao renovar os créditos a cada dois créditos consumidos por um cliente, o dispositivo mestre faz uso de um pacote de sinalização. Ou seja, a cada três pacotes trocados no canal de transmissão, um pacote é de sinalização.

Inicialmente, isso parece ser um comportamento indesejado, entretanto, com um número menor para a renovação de créditos o dispositivo mestre consegue ter um controle maior do cliente. Por exemplo, se um dispositivo mestre sempre renovar os créditos um-a-um ao cliente, isso faz com que o mesmo tenha o controle de quando o cliente pode enviar ou não pacotes. Isso se faz possível dado que o cliente vai ter que esperar um novo crédito para o envio de um novo pacote. Portanto, o número padrão para a troca de créditos deve ser escolhido a depender do uso e nível de controle que o dispositivo mestre deseja ter em relação aos clientes.

Além da avaliação da taxa de transmissão máxima fim-a-fim também foram realizados experimentos para avaliação da taxa de transmissão máxima suportada em um dispositivo mestre com múltiplos clientes. O objetivo principal foi avaliar se a comunicação com múltiplos clientes influenciaria no funcionamento do controlador Bluetooth.

De maneira semelhante ao anterior, esse experimento foi realizado de maneira incremental. Inicialmente foi conectado um dispositivo cliente com uma taxa constante. Em seguida outros clientes foram conectados sequencialmente até que a conexão fosse abortada pelo dispositivo mestre. O gráfico da Figura 5.4 apresenta os resultados de um experimento realizado com o controlador Broadcom. Nesse experimento dois clientes foram conectados e tiveram sua conexão iniciada, os quais são representados pelos gráficos em cor $azul$ e $vermelha$. Após essas duas conexões, o dispositivo mestre já se encontrava próximo de sua taxa de transmissão máxima. Depois de $60$ segundos de comunicação com esses dois clientes, um terceiro cliente foi conectado, nesse instante, representado pelo ponto $(a)$ no gráfico, o dispositivo mestre não foi mais capaz de manter todas as conexões e abortou a comunicação com todos os clientes.

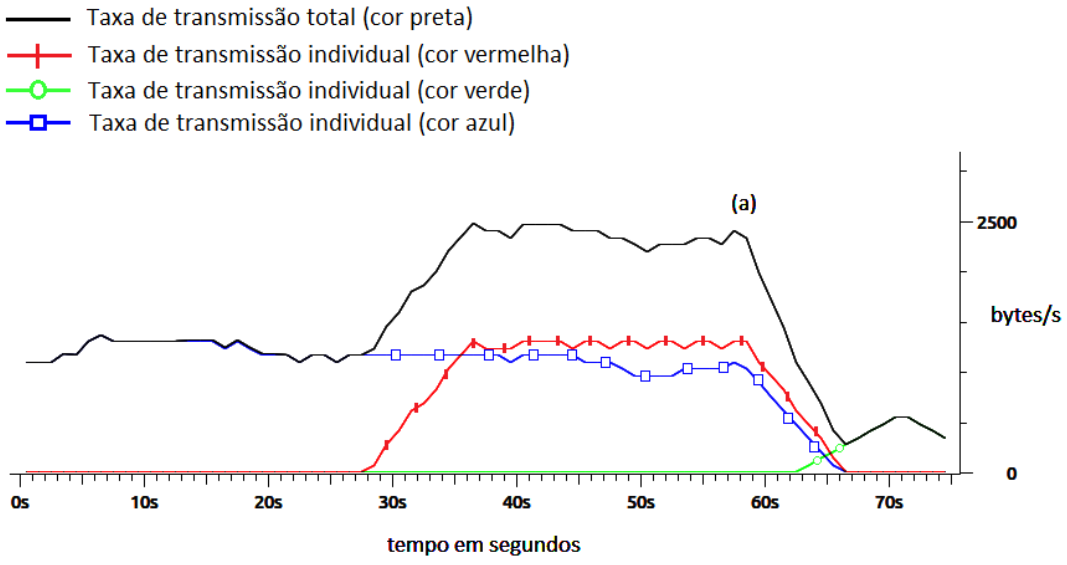


Figura 5.4: Taxa de transmissão máxima de dados total em um nó mestre com chipset Broadcom BCM20702 e três clientes.

Foi observado que a taxa de transmissão máxima suportada no dispositivo cliente não aparenta ser influenciada pelo número de clientes conectados, e sim, pelo número de comandos de créditos trocados durante a comunicação. Por exemplo, quanto mais clientes

conectados, mais comandos de créditos serão trocados mesmo que todos os clientes façam uso de um valor padrão de troca de créditos altos.

Por fim, na avaliação do impacto na taxa de transmissão com múltiplos clientes, apenas o controlador Broadcom foi utilizado como dispositivo mestre. Isso se deve ao fato dele ser o único a compartilhar a banda de transmissão em nível de enlace de maneira homogênea entre todos os clientes. Os controladores da Intel e da CSR priorizam sempre o último dispositivo cliente conectado, disponibilizando para esse uma taxa de transmissão maior. Eles fazem isso através do algoritmo de controle de acesso ao meio em nível de enlace. Não foi possível aferir se isso é um defeito ou uma decisão de projeto desses controladores.

Portanto, para a realização dos experimentos no próximo capítulo foi utilizado o controlador Broadcom como dispositivo mestre, dado que este não executa nenhum tipo de algoritmo de priorização, ou apresenta defeito no controle de acesso ao meio na camada de enlace.

## 5.2 Projeto do Controlador Adaptativo

Com o problema de limitação dos controladores Bluetooth exposto, neste trabalho é proposto a criação de um novo controlador de fluxo de dados para o Bluetooth Low-Energy. Esse novo controlador faz uso do novo mecanismo de controle de fluxo baseado em créditos introduzido no Bluetooth 4.2 e descrito no Capítulo 2. Tendo como objetivo evitar que o controlador do dispositivo mestre entre em um estado de funcionamento instável e pare de funcionar, no projeto de desenvolvimento desse novo controlador alguns requisitos básicos foram definidos:

- O dispositivo mestre conhece a priori a limitação de banda de seu controlador Bluetooth.
- O dispositivo mestre deve controlar o fluxo de dados originado por suas aplicações e serviços.
- O dispositivo mestre da rede deve controlar o fluxo de dados de seus clientes (*slaves*) quando necessário através da alocação dinâmica de créditos para os mesmos.

O funcionamento de um típico controlador baseado em créditos para o Bluetooth Low-Energy é apresentado na Figura 5.5. Em uma rede BLE com modo de controle de fluxo

baseado em créditos, o dispositivo mestre faz o controle de dados na camada de enlace no controlador, e na camada L2CAP no lado hospedeiro da pilha de protocolos Bluetooth.

Seguindo o fluxo apresentado na Figura 5.5, o controle de créditos é executado toda vez que o cliente envia um novo pacote *LE-Frame* ao mestre. Ao receber um novo *LE-Frame*, o dispositivo mestre decrementa a contagem de créditos daquele cliente e verifica qual a contagem atual do mesmo. Caso essa contagem fique menor do que um valor $X$, o dispositivo mestre decide enviar mais $Y$ créditos ao dispositivo cliente, fazendo que o mesmo fique com $X + Y$ créditos disponíveis para envio, onde $X$ e $Y$ são parâmetros de implementação do L2CAP do mestre.

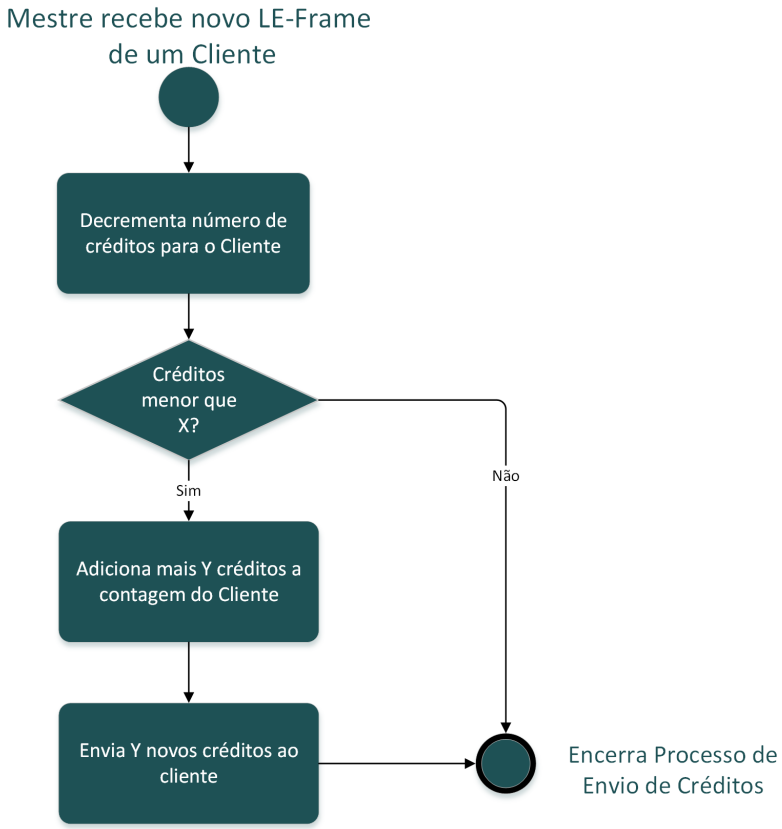


Figura 5.5: Fluxo de controle baseado em créditos padrão.

Nesse modelo padrão de controle de distribuição de créditos, todos os clientes sempre têm créditos disponíveis para envio, portanto, esses dispositivos tem liberdade para enviar dados a todo o momento que *slots* de tempo na camada de enlace estejam disponíveis. Considerando, como apresentado na seção anterior, que o controlador Bluetooth é limitado em termos de processamento e memória, além de não implementar controles de acesso sofis-

ticados na camada de enlace, o modelo de distribuição homogênea de créditos na camada L2CAP faz com que todos os clientes tenham o mesmo direito de acesso ao canal. Portanto, todos os clientes têm a seu dispor canais sem garantia de QoS, ou seja, canais *best-efforts*.

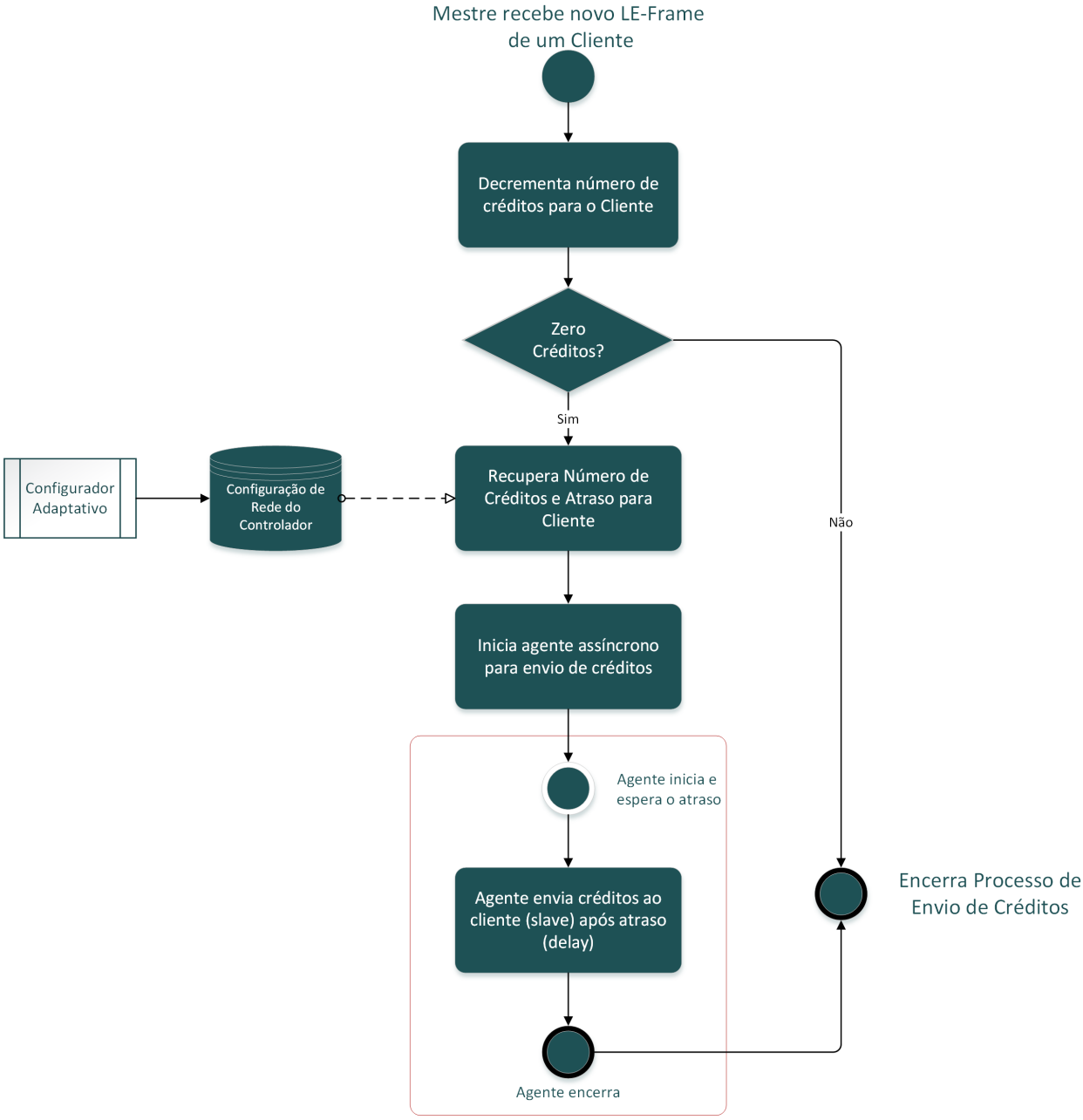


Figura 5.6: Fluxo de controle baseado em créditos com regras e agentes de distribuição.

Á vista disso, nesse trabalho é proposto um novo modelo de controle de fluxo baseado em créditos, o qual faz uso de regras de controle através de um *Configurador Adaptativo*, e de *Agentes* independentes de distribuição de créditos. Esse novo modelo de controle é

apresentado na Figura 5.11.

De maneira semelhante ao controlador típico da Figura 5.5, ao receber um novo *LE-Frame* de um cliente o dispositivo mestre decrementa a contagem de créditos daquele nó. Nesse ponto, o novo controlador verifica se a contagem de créditos do cliente alcançou o valor $zero$ ou não. O valor $zero$ foi escolhido para dá a oportunidade ao dispositivo mestre de "segurar" o envio de novos *LE-Frames* por parte daquele cliente. Isso permite que o dispositivo mestre controle o envio de novos pacotes por parte do cliente através de uma estratégia de "pressão reversa" ou *backpressure*. Essa estratégia faz com que as aplicações e serviços nos clientes recebam alertas informando que o meio não está disponível naquele momento, com isso, permitindo que as mesmas sejam capazes de lidar com tal situação antes do envio de novos dados ao meio.

Através dessa definição de estratégia de controle, o novo controlador proposto faz uso de regras de controle geradas e armazenadas por um *Configurador Adaptativo*. Essas regras informam os seguintes parâmetros ao controlador:

- Número de créditos que deve ser retornado ao cliente, $Cr_n$.
- Tempo de atraso para o envio desses créditos ao cliente, $Del_n$.

Esses dois parâmetros são calculados pelo *Configurador Adaptativo* a partir de uma definição de canal BLE baseado em créditos, onde:

**Definição 4 (Modelo de um Canal BLE Baseado em Créditos)** .

*Canal BLE L2CAP em créditos:*

$$Canal_{BLE} = \{Td_n, Cr_n\}$$

*Com isso, a taxa instantânea máxima de transmissão é:*

$$Tx_n = Cr_n * MTU/Del_n$$

*Onde:*

$Tx_n$ *é a taxa de transmissão em* $bytes/s$.

$MTU$ *é o tamanho máximo de um LE-Frame.*

$Del_n$ *é o tempo para envio de* $Cr_n$ *em segundos.*

Com isso, ao receber esses parâmetros para um dispositivo identificado por $DeviceId_n$, o *Agente* apresentado na Figura 5.5 atua de maneira assíncrona, esperando um período de tempo $Del_n$ para o envio de $Cr_n$ créditos ao cliente. Se considerarmos que o cliente irá utilizar os $Cr_n$ imediatamente depois recebe-lo, o modelo de canal apresentado na Definição 4 se aplica, e a taxa de transmissão máxima permitida $Tx_n$ é alcançada.

Esse modelo generalizado para controle de fluxo baseado em créditos torna-se base para a aplicação de diferentes modelos de controle executados pelo *Configurador Adaptativo*. Os seguintes modelos de controle foram estudados e aplicados ao controlador adaptativo:

- Modelo de controle com uma distribuição simples de créditos entre os clientes.
- Modelo de controle com uma distribuição de créditos baseada em características de QoS de cada cliente.
- Modelo de controle com uma distribuição de créditos baseada em características de QoS e na prioridade de cada cliente na rede.
- Modelo de controle com uma distribuição de créditos baseada em características de QoS e na prioridade temporária de cada cliente na rede.

Esses modelos foram desenvolvidos e avaliados de maneira incremental. Nesse sentido, as seções a seguir apresentam o processo evolutivo de projeto e desenvolvimento do controlador adaptativo baseado em créditos para redes BLE.

### 5.2.1 Parâmetros de Entrada e Configuração do Controlador

Para ser capaz de adaptar-se as características e limitações impostas pelo dispositivo BLE, o controlador adaptativo baseado em créditos necessita de parâmetros de entrada e configuração. Dada a limitação de dispositivos controladores BLE apresentada em seções anteriores, o controlador adaptativo precisa ter ciência dos seguintes parâmetros de entrada:

- $maxTx$, a taxa máxima de dados que o dispositivo controlador BLE suporta.
- $MTU$, o tamanho máximo em $bytes$ de cada *LE-Frame* utilizado pelo nó mestre.
- $N$, o número de nós clientes conectados.

Além desses parâmetros de entrada, o controlador faz uso de um parâmetro de configuração, definido como:

- $xCrs$, o número mínimo de créditos que devem ser enviados a um cliente por interação.

As mensagens com as informações de controle para a alocação de novos créditos entre os dispositivos fazem uso do mesmo canal de troca de dados. Ou seja, *LE-Frames* são utilizados para o envio de novos créditos a um cliente. Com isso, cada vez que um dispositivo envia mais créditos a outro, esses deixam de utilizar o canal para a troca de dados, portanto, diminuindo a taxa de utilização desse canal para dados, e consequentemente, diminuindo a taxa de transmissão máxima alcançada no canal.

Por conseguinte, um valor $xCrs$ ótimo deve ser encontrado para cada dispositivo controlador BLE, de modo que a taxa de transmissão de dados com esse parâmetro se aproxime do valor $maxTx$. Na implementação e avaliação experimental dos modelos de controle apresentado no Capítulo 6, também foram realizados experimentos para a escolha de um valor ideal de $xCrs$.

### 5.2.2 Controle com Distribuição Simples de Créditos

Com o objetivo de evitar que a soma da taxa de transmissão utilizada por todos os clientes de um mestre ultrapasse o limite $maxTx$, o modo de controle mais simples é dividir $maxTx$ igualmente entre todos os dispositivos conectados. Nesse modelo de controle simples, apenas os parâmetros de entrada apresentados na seção anterior são utilizados.

O processo de funcionamento do *Configurador Adaptativo* pode ser descrito através do diagrama da Figura 5.7. O processo é executado e atualizado toda vez que um evento de conexão ou desconexão de um novo dispositivo cliente é identificado. Ao receber esse evento, o *Configurador Adaptativo* executa o algoritmo de distribuição simples de créditos (SCD). Esse algoritmo tem como entrada os parâmetros $maxTx$, $MTU$ e $xCrs$, além da lista de dispositivos conectados $AllNodes$. A saída do algoritmo é uma lista para cada cliente $n$ com o valor da taxa de transmissão $nTx$, período de atraso para o envio de créditos $Del[n]$ e o número de créditos para enviar a cada interação $Crs[n]$. Esses valores são armazenados no controlador de rede, e utilizados pelo controlador de fluxo apresentado na Figura 5.11.

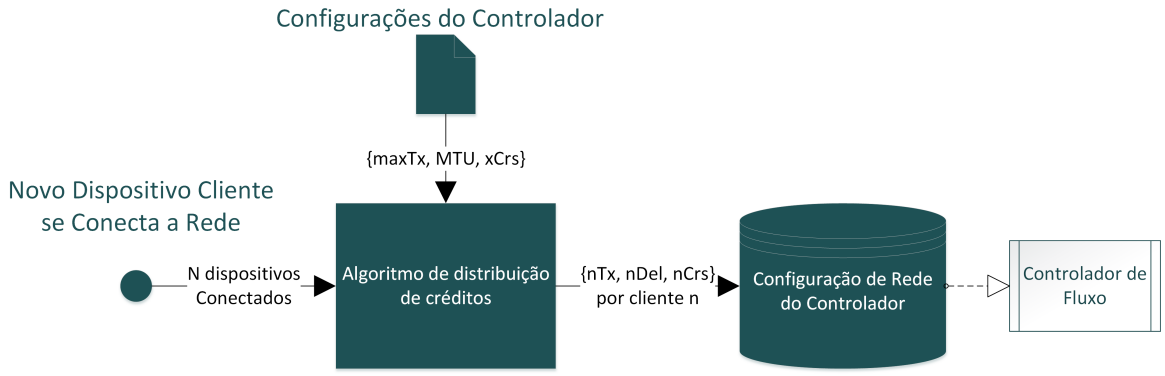


Figura 5.7: Fluxo de criação de regras de controle simples no Configurador Adaptativo.

O algoritmo SCD pode ser descrito através do pseudocódigo apresentado no Código Fonte 5.1. Ao ser executado, o SCD calcula inicialmente o número total de nós conectados $N$ (linha 2), e define a taxa máxima que cada dispositivo tem direito $sTx$ (linha 3). Com esses valores, e considerando o valor mínimo de créditos $xCrs$ que devem ser enviados por interação a cada dispositivo, o SCD calcula o tempo de atraso $sDel$ que deve ser aplicado para o envio desses créditos a cada dispositivo (linha 4). O valor de $sDel$ é calculado como sendo a divisão do número total de bytes que devem ser enviados por interação $xCrs * MTU$ dividido pela taxa $sTx$ que cada dispositivo tem direito.

Código Fonte 5.1: Pseudo Código do Algoritmo SCD

```
1 SCD(maxTx, xCrs, MTU, AllNodes)
2 N := sizeOf(AllNodes)
3 sTx := maxTx/N
4 sDel := (xCrs*MTU) / sTx
5 for each node n in AllNodes:
6 Tx[n] := sTx
7 Del[n] := sDel
8 Crs[n] := xCrs
9 end for
```

Após os cálculos de $sTx$ e $sDel$, esses valores são aplicados a configuração individual de cada dispositivo $n$ e armazenados no *Configurador Adaptativo* para uso dos *Agentes* de distribuição de créditos (linhas 5-9).

### 5.2.3 Controle com Distribuição de Créditos Baseado em QoS

Apesar de manter a taxa de transmissão máxima dentro do limite $maxTx$, o controle através do algoritmo SCD não faz uso das características de cada dispositivo. Por exemplo, utilizando o SCD o controlador adaptativo pode alocar uma taxa $sTx$ para um dispositivo, entretanto, esse dispositivo apenas necessita de metade desse valor. Por outro lado, outro dispositivo pode necessitar de uma taxa maior que $sTx$. Nesse sentido, ter ciência e fazer uso de características de transmissão de cada dispositivo torna-se uma alternativa interessante para a distribuição de créditos de maneira equilibrada entre os nós.

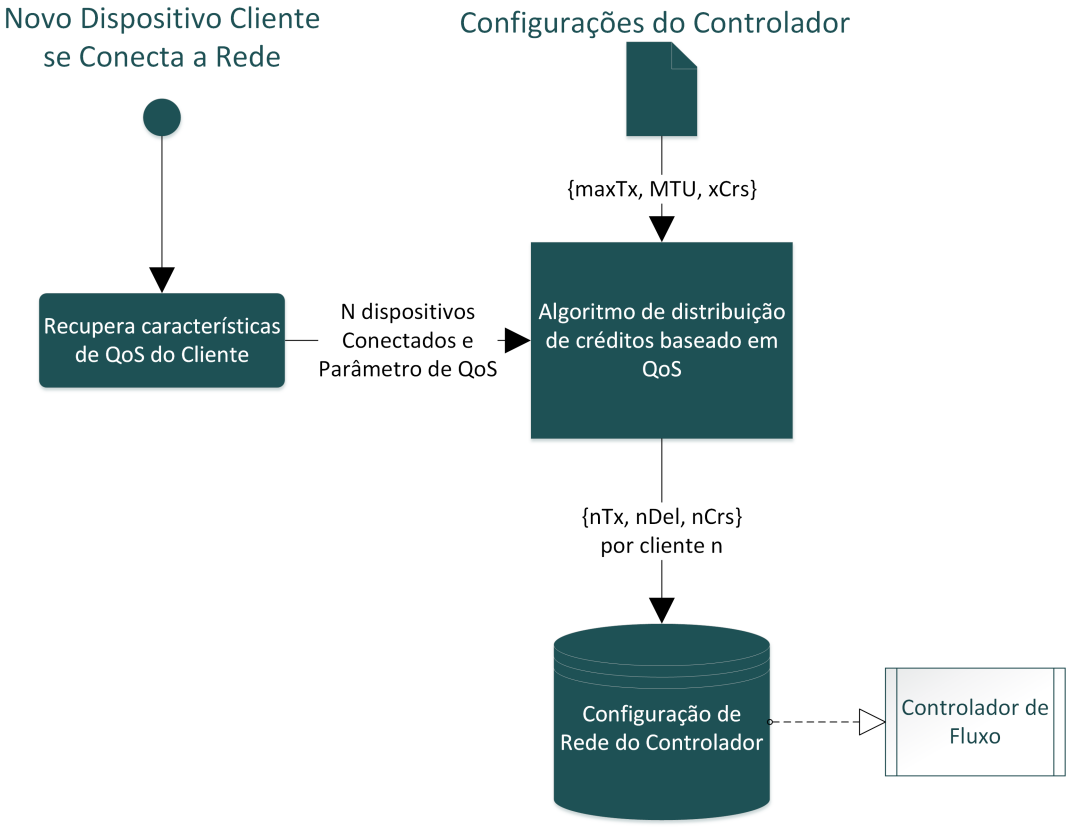


Figura 5.8: Fluxo de criação de regras de controle baseado em QoS.

Com isso, um novo modelo de controle é utilizado onde características de QoS de cada dispositivo são utilizadas na distribuição de créditos. O diagrama da Figura 5.8 apresenta o fluxo de distribuição de créditos com o uso de características de QoS de cada cliente. A principal diferença desse modelo em relação ao anterior é o passo de recuperação de características de QoS do cliente. Quando um novo dispositivo conecta-se a rede, o Controlador

Adaptativo recupera suas características de QoS através de um processo paralelo de leitura de características através do protocolo GATT BLE, o qual é descrito na Seção 5.2.6. Na versão proposta nesse trabalho, o parâmetro de QoS que descreve o dispositivo se caracteriza pela sua taxa desejada de transmissão $TxD$.

Com os parâmetros de QoS incluído na lista de todos os dispositivos conectados, o *Configurador Adaptativo* executa o algoritmo de distribuição de créditos QADC descrito no Código Fonte 5.2. O princípio do algoritmo QADC consiste em distribuir os créditos inicialmente com os dispositivos com parâmetros de QoS, considerando a taxa ideal $TxD$ de cada um. Após essa distribuição, os créditos são distribuídos de maneira igualitária entre os dispositivos sem parâmetros de QoS seguindo o algoritmo SCD.

Outra característica de configuração utilizada por esse controlador é a definição de uma taxa de transmissão mínima $minFreeTx$ que deve ser garantida para a distribuição entre os nós sem parâmetro de QoS. Esse parâmetro foi definido para evitar que os dispositivos com parâmetros de QoS consumam toda banda de transmissão disponível. Por ser um parâmetro configurável, $minFreeTx$ pode ser configurado ao valor zero, portanto, permitindo que os dispositivos com parâmetros de QoS consumam toda banda quando necessário.

Com as definições apresentadas, o algoritmo QACD inicia com a separação dos dispositivos em duas listas, com e sem parâmetro de QoS (linhas 2-8). Após essa separação, é definido o parâmetro $availableTx$ como sendo a taxa de transmissão disponível para divisão naquele momento (linha 10).

Com essa definição, é realizada uma interação na lista de dispositivos com parâmetros de QoS $qosList$ (linha 12). Seguindo a ordem de prioridade na lista $qosList$, a taxa de transmissão disponível $availableTx$ é decrescida do valor $TxD$ de cada dispositivo $n$ (linha 13). Após cada decréscimo no valor de $availableTx$ é realizada a checagem se o valor $minFreeTx$ foi atingido (linha 14). Em caso positivo, a interação é abortada, e todos os dispositivos pertencentes a lista $qosList$ são inseridos na lista $normalList$, portanto, seus parâmetros de QoS são ignorados (linha 17). Caso negativo, a interação contínua e são realizados os cálculos do número de créditos $Crs$, e do período de atraso $Del$ para cada dispositivo $n$ semelhantemente ao apresentado no algoritmo SCD (linha 23).

Por fim, após a distribuição de créditos aos dispositivos com parâmetros de QoS, os nós restantes na lista $normalList$ tem sua distribuição de créditos e período de atraso calculados

a partir do algoritmo SCD. É importante observar que para a execução do SCD são considerados a taxa de transmissão remanescente $availableTx$ e a lista de dispositivos $normalList$.

Código Fonte 5.2: Pseudo Código do Algoritmo QACD

```
1  QACD(maxTx, xCrs, MTU, AllNodes)
2  for each node n in AllNodes:
3  if n has qosParam:
4  push(qosList, n)
5  else
6  push(normalList, n)
7  end if
8  end for
9
10 availableTx := maxTx
11
12 for each node q in qosList:
13 availableTx := availableTx- TxD[n]
14 if availableTx < minFreeTx:
15 normalList := allNodes
16 availableThr := maxThr
17 end for
18 end if
19 Del[n] := (xCrs*MTU) / TxD[n]
20 Crs[n] := xCrs
21 end for
22
23 SCD(availableTx, xCrs, MTU, normalList)
```

Semelhantemente ao modelo de distribuição simples, os valores de $sTx$ e $sDel$, são aplicados a configuração individual de cada dispositivo $n$ e armazenados no *Configurador Adaptativo* para uso pelos *Agentes* de distribuição de créditos.

### 5.2.4 Controle com Distribuição de Créditos Baseado em QoS com Prioridade

Uma das limitações do modelo de controle definido anteriormente ocorre quando todos os dispositivos na rede tem parâmetros de QoS. Nesse caso específico, os recursos de rede (taxa

de transmissão) são distribuídos inicialmente com os primeiros dispositivos na lista $qosList$, podendo levar a uma situação onde a taxa disponível $availableTx$ não seja suficiente para os requisitos de QoS de alguns nós.

Para evitar situações onde dispositivos com prioridade sejam prejudicados devido a sua distribuição em uma lista, um novo modelo de distribuição é proposto, onde uma lista com prioridade entre dispositivos é fornecida por aplicações. Esse novo modelo é apresentado no diagrama da Figura 5.9. Nesse novo modelo, a cada novo dispositivo conectado, além de recuperar seus parâmetros de QoS através da leitura de características GATT, o *Configurador Adaptativo* verifica se esse novo dispositivo tem algum tipo de prioridade para as aplicações do sistema. Prioridade é representada por um inteiro positivo, onde o menor valor indica a maior prioridade.

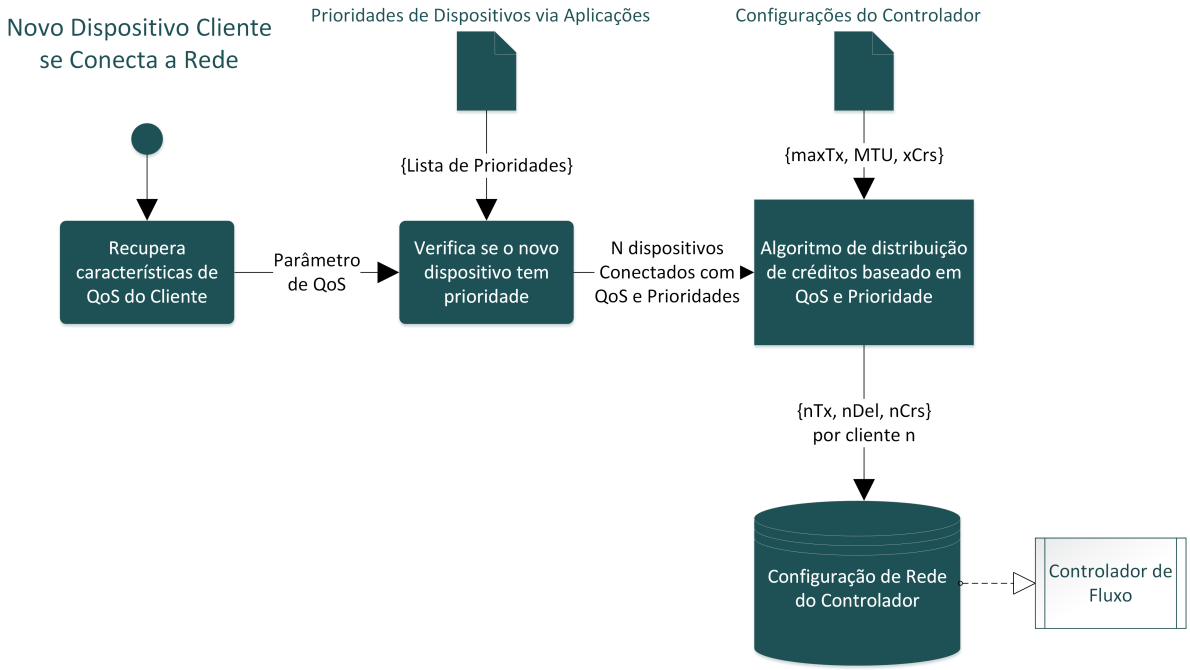


Figura 5.9: Fluxo de criação de regras de controle baseado em QoS com Prioridade.

Após essa atualização do valor de prioridade $prio$ e do parâmetro de QoS $TxD$ de cada dispositivo, o algoritmo de distribuição de créditos PQACD é executado. O algoritmo PQACD é apresentado no Código Fonte 5.3. Esse algoritmo tem o funcionamento semelhante ao algoritmo QACD, sendo a única diferença presente na inserção dos dispositivos com parâmetros de QoS na lista $qosList$ (linha 4). Nesse novo algoritmo, a cada inserção de um nó $n$ na lista $qosList$, o valor de prioridade $prio$ é inserido como indexador daquele

dispositivo na lista. Com isso, ao realizar a interação na lista $qosList$ para a distribuição de créditos, os dispositivos com valor de $prio$ menores serão priorizados.

Código Fonte 5.3: Pseudo Código do Algoritmo PQACD

```
1  PQACD(maxTx, xCrs, MTU, AllNodes)
2  for each node n in allNodes:
3  if n has qosParam:
4  insert(qosList, n, prio[n])
5  else
6  push(normalList, n)
7  end if
8  end for
9
10 availableTx := maxTx
11
12 for each node q in qosList:
13 availableTx := availableTx- TxD[n]
14 if availableTx < minFreeTx:
15 normalList := normalList
16 availableThr := maxThr
17 end for
18 end if
19 Del[n] := (xCrs*MTU) / TxD[n]
20 Crs[n] := xCrs
21 end for
22
23 SCD(availableTx, xCrs, MTU, normalList)
```

Em algumas situações, é importante ressaltar que o uso do algoritmo PQACD só é eficaz caso o valor de $minFreeTx$ seja nulo. Ou seja, ele apenas é útil caso não se reserve uma banda de transmissão mínima aos dispositivos sem parâmetro de QoS. Por exemplo, caso um dispositivo $n$ com parâmetro de QoS $TxD$ tenha a maior prioridade, e $TxD[n] > availableTx - minFreeTx$, esse dispositivo nunca vai ser capaz de ter uma garantia de requisitos para o valor $TxD$, dado que a checagem de valor $minFreeTx$ é infringida.

Com isso, caso o uso de prioridades seja essencial ao funcionamento de aplicações do

sistema que faz uso do controlador adaptativo para BLE, o valor de $minFreeTx$ deve ser nulo, garantido assim o funcionamento dos nós com maior prioridade em todas as situações. Esse é o caso do Sistema de Monitoramento Remoto de Paciente apresentado no Capítulo 4.

### 5.2.5 Controle com Distribuição de Créditos Baseado em QoS com Prioridade Temporal

Além da prioridade predefinida que um dispositivo pode ter em relação a outro, um aspecto de prioridade temporal pode ser adicionado. Isso significa que um dispositivo não precisa ter necessariamente uma prioridade sobre outro indefinidamente. Por exemplo, pode-se definir que um dispositivo vai ter prioridade sobre outros dispositivos apenas durante $Ps$ segundos após o inicio da transmissão. Esses casos se aplicam quando aplicações já têm ciência do tipo do dispositivo e seu uso. Como exemplo, uma aplicação de controle pode definir que um dispositivo do tipo $ECG$ deve sempre ter prioridade durante os três primeiros minutos de uso. A Figura 5.10 apresenta o diagrama com o fluxo de criação de regras.

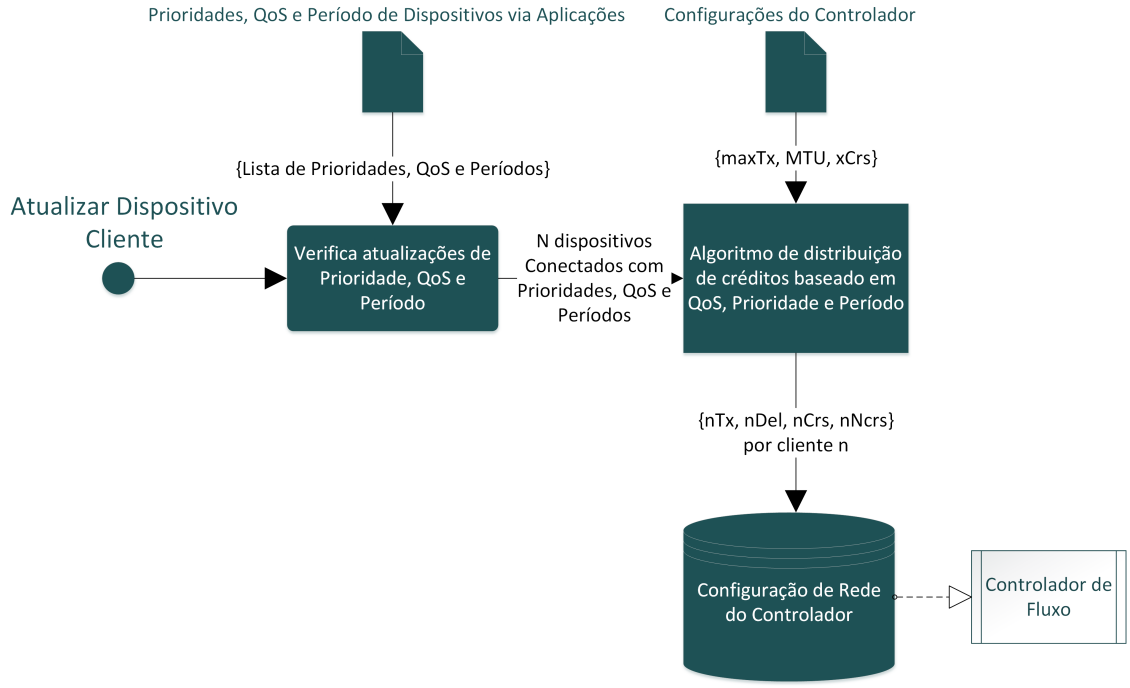


Figura 5.10: Fluxo de criação de regras de controle baseado em QoS com Prioridade Temporal.

A principal diferença em relação ao modelo anterior está no envio do período de tempo

$Ps[n]$ pelas aplicações de controle para certos dispositivos, e pelo cálculo do número total de créditos $Ncrs[n]$ que devem ser enviados a um dispositivo $n$. Esse processo de atualização de regras é executado toda vez que for necessário atualizar um dispositivo cliente, por exemplo, por comando de uma aplicação ou serviço do dispositivo hospedeiro. Com esses novos parâmetros, o algoritmo de distribuição de créditos TPQACD, apresentado no Código Fonte 5.4, é executado.

Código Fonte 5.4: Pseudo Código do Algoritmo TPQACD

```
1  TPQACD(maxTx, xCrs, MTU, AllNodes)
2  for each node n in allNodes:
3  if n has qosParam:
4  insert(qosList, n, prio[n])
5  else
6  push(normalList, n)
7  end if
8  end for
9
10 availableTx := maxTx
11
12 for each node q in qosList:
13 availableTx := availableTx- TxD[n]
14 if availableTx < minFreeTx:
15 normalList := normalList
16 availableThr := maxThr
17 end for
18 end if
19 Del[n] := (xCrs*MTU) / TxD[n]
20 Crs[n] := xCrs
21 Ncrs[n] := Crs[n]*(Ps[n] / Del[n])
22 if Ncrs[n] is 0:
23 Ncrs[n] := -1
24 end if
25 end for
26
27 SCD(availableTx, xCrs, MTU, normalList)
```

O algoritmo apresentado no Código Fonte 5.4 tem como diferencial em relação ao algo-

ritmo PQACD o cálculo do número máximo de créditos $Ncrs[n]$ que devem ser alocados a um dispositivo (linha 21). Esse número é calculado considerando o valor o período de tempo $Ps[n]$ e o atraso entre interações de créditos $Del[n]$ que foi calculado.

Algumas questões adicionais devem ser consideradas com o uso de prioridade temporal:

- Como definir um período indeterminado de prioridade para um dispositivo?
- O que acontece quando o período de prioridade de um dispositivo encerra?
- Como controlar a mudança de prioridade de um dispositivo ao final desse período?

Em resposta a primeira pergunta foi definido que quando o período de tempo for indeterminado, o controlador vai aplicar o valor nulo ao período $Ps[n]$ do dispositivo. Isso irá anular o número total de créditos $Ncrs[n]$ calculado, o que vai defini-lo com um valor negativo como apresentado no algoritmo no Código Fonte 5.4 (linhas 22-24).

Em relação a segunda pergunta foi definido que ao encerrar o número total de créditos $Ncrs[n]$ enviados a um dispositivo $n$, esse dispositivo adquire prioridade padrão. Prioridade padrão é a mesma adquirida por dispositivos que não tem a prioridade definida por aplicações, ou seja, tenha a taxa de transmissão $TxD$ calculada através do algoritmo SCD. Por fim, para controlar essas mudanças de prioridade foi inserido um Controlador de Interações (CI) ao Controlador Adaptativo, como ilustrado no diagrama da Figura 5.11.

O Controlador de Interações funciona como um contador de créditos disponíveis para cada dispositivo. Toda vez que o Controlador Adaptativo requisita o número de créditos disponíveis para um dispositivo, o CI decrementa a contagem de créditos máximo $Ncrs[n]$ para aquele dispositivo. Caso $Ncrs[n]$ seja negativo, ou seja, o dispositivo $n$ tem prioridade por tempo indeterminado, o CI sempre irá retornar o valor $Crs[n]$ e $Del[n]$ com prioridade para esse dispositivo. Caso o $Ncrs[n]$ seja positivo, o CI irá retornar e decrementar o valor de créditos $Crs[n]$ com prioridade a cada interação, até que o valor de $Ncrs[n]$ seja nulo. Após o valor ser anulado, o CI começa a retornar para esse dispositivo o valor de créditos $Crs[n]$ e atraso $Del[n]$ padrão para dispositivos sem prioridade.

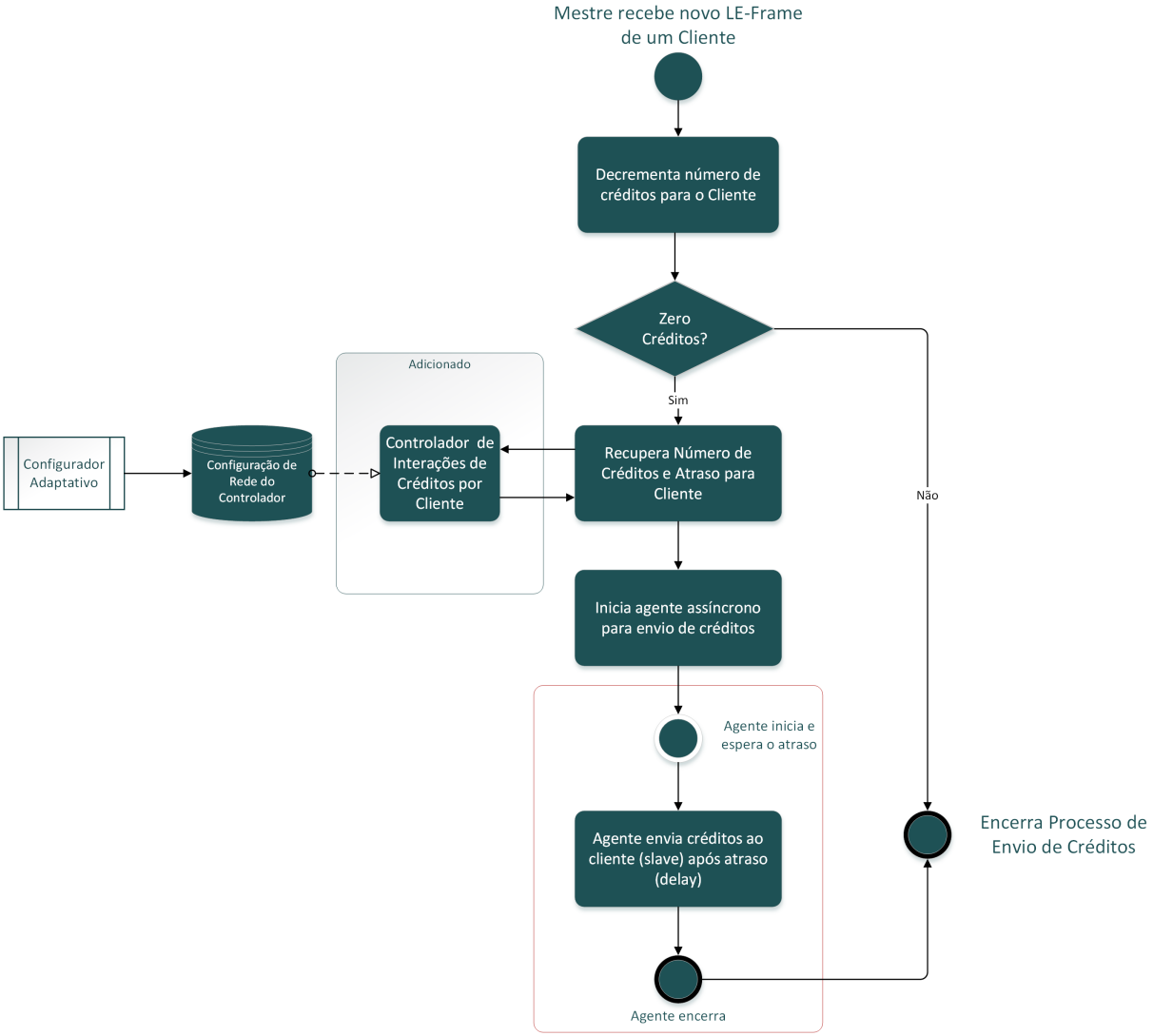


Figura 5.11: Fluxo de controle baseado em créditos com regras e agentes de distribuição e controle de interações.

### 5.2.6 Descritor de Dispositivo para Controle Baseado em QoS

Para auxiliar na identificação dos parâmetros de QoS de cada dispositivo, foi definido um serviço GATT para Bluetooth Low-Energy que deve ser instanciado por cada dispositivo que deseje ter seus parâmetros de QoS lidos. O serviço chamado de *QoS Descriptor* oferece uma característica obrigatória chamada de *Standard Throughput*. Essa característica é de apenas leitura e retorna a banda de transmissão necessária para o correto funcionamento do dispositivo $TxD$. Essa banda de transmissão desejada é utilizada como parâmetro de QoS nos modelos de controle apresentados anteriormente.

Portanto, além de implementar o perfil IPSP como apresentado no Capítulo 2, os dispositivos participantes de uma rede BLE com suporte a IPv6 e Controle Adaptativo de Fluxo podem implementar o serviço GATT *QoS Descriptor*. Esse serviço é opcional, portanto, caso não seja implementado por um dispositivo cliente, este é considerado como um dispositivo sem requisitos de QoS. O fluxo de funcionamento de leitura de características de QoS é apresentado no Diagrama da Figura 5.12.

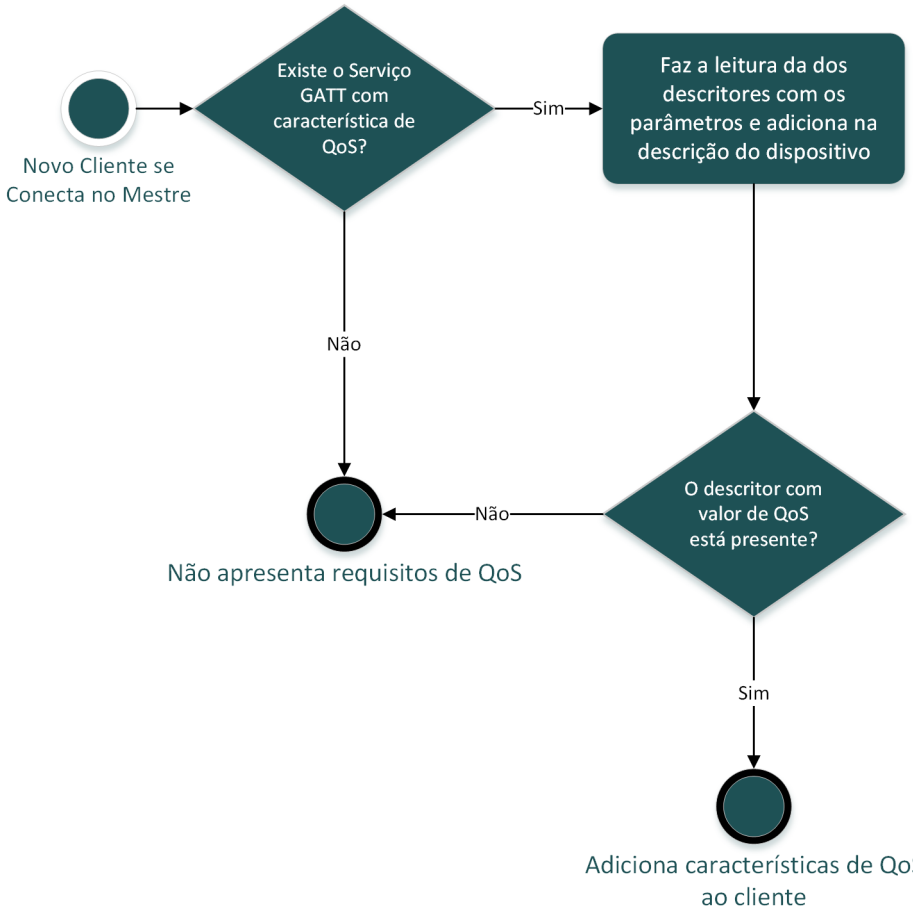


Figura 5.12: Diagrama com processo de leitura de características de QoS de um dispositivo.

## 5.3 Considerações Finais do Capítulo

Neste capítulo foi apresentado um modelo de controle adaptativo de fluxo de dados para o Bluetooth Low-Energy com suporte IPv6. Esse modelo de controle tem como ponto central o controlador BLE do dispositivo mestre, o qual é responsável pela formação da rede BLE. Foi apresentado o projeto de referência desse controlador adaptativo, além de diferentes modelos

de controle que podem ser instanciados e utilizados. Antes do desenvolvimento do projeto de referência do controlador, foi discutida a motivação para o seu desenvolvimento, assim como apresentados resultados experimentais que exemplificam a limitação de controladores Bluetooth na prática.

Para o projeto dos modelos de controle foi realizado um trabalho de desenvolvimento incremental. Esse trabalho consistiu em desenvolver um controlador mais simples e ir evoluindo-o através de necessidades específicas que foram identificadas no desenvolvimento do sistema de Monitoramento Remoto de Pacientes apresentado no Capítulo 4. Esses modelos de controles foram descritos através de fluxogramas e algoritmos de distribuição de créditos.

# Capítulo 6

# Avaliação Experimental do Controlador Adaptativo para Gateways Bluetooth Low-Energy

Como detalhado no Capítulo 5, é possível realizar um controle adaptativo de fluxo de dados em Gateways IPv6 Bluetooth Low-Energy (BLE) através de uma distribuição inteligente de créditos entre os clientes. Neste Capítulo, portanto, são apresentados detalhes sobre o processo de desenvolvimento do software do controlador adaptativo de fluxo para Gateways BLE. Além desses detalhes de implementação, são descritos os procedimentos experimentais realizados para avaliação e verificação do comportamento do controlador em diferentes situações. O objetivo dos experimentos realizados neste capítulo é avaliar, independentemente de aplicação, a funcionalidade de controle de fluxo baseado em créditos em Gateways BLE. Após a verificação dos resultados do Gateway de maneira independente, será realizada a integração do mesmo ao Sistema de Monitoramento Remoto de Pacientes descrito no Capítulo 4. A descrição e os resultados dessa integração são apresentados no Capítulo 7.

## 6.1 Desenvolvimento de um Protótipo para Avaliação

Para o desenvolvimento do protótipo de um controlador adaptativo para Gateways IPv6 BLE foi escolhido o sistema operacional Linux como ambiente de testes. A versão alvo do Kernel do Linux escolhida foi a 4.2.0-17, com uma distribuição Ubuntu 15.10.

De maneira simplificada, no que condiz a funcionalidade do Bluetooth, o Kernel do Linux apresenta um módulo independente para o subsistema Bluetooth, como ilustrado na Figura 6.1. Dentro desse módulo Bluetooth, existem outros submódulos relativos a diferentes camadas e protocolos da especificação Bluetooth. Relativos ao controle de fluxo para redes BLE com IPv6, dois módulos destacam-se:

- Módulo Controlador da camada L2CAP, representado na Figura 6.1(b).
- Módulo Controlador do 6LoWPAN Bluetooth, representado na Figura 6.1c.

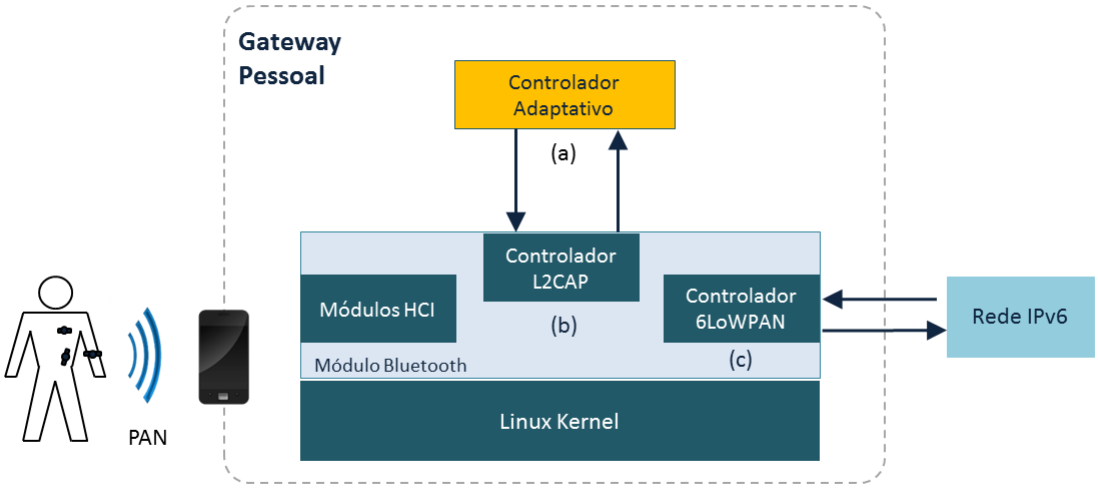


Figura 6.1: Diagrama com modelo de implementação do controlador adaptativo.

Dentre as funcionalidades do módulo L2CAP, no que tange o controlador apresentado nesse trabalho, destaca-se a funcionalidade do controle de créditos entre os clientes conectados a um Gateway BLE. Portanto, é esse módulo que define como vai ser a distribuição e controle de créditos. O módulo 6LoWPAN, por sua vez, fica responsável pela interação e transporte de pacotes entre a camada L2CAP e camada de rede relativa ao IPv6 no sistema operacional Linux.

Antes de realizar a implementação do controlador adaptativo e seus algoritmos de controle, foi necessário realizar alterações nesses dois módulos citados. Considerando as funcionalidades do controlador adaptativo, os seguintes requisitos básicos foram identificados, os quais são dependentes dos módulos Bluetooth do sistema operacional:

1. O controlador adaptativo precisa ter ciência quando um cliente se conecta e desconecta a rede.

2. O controlador adaptativo precisa ter uma identificação única para cada cliente conectado.

3. O controlador adaptativo precisa ter ciência do número de créditos alocados para cada cliente.

4. O controlador adaptativo precisa ser capaz de comandar o envio de créditos adicionais aos clientes de maneira independente.

Para viabilizar que esses requisitos sejam atendidos, foi necessário implementar dois mecanismos básicos de entrada e saída de comandos entre uma aplicação externa, o controlador adaptativo, e os submódulos Bluetooth do Kernel, os quais são representados na Figura 6.1(a).

Para a saída das informações necessárias aos requisitos *1, 2 e 3* foram realizadas alterações nos módulos L2CAP e 6LoWPAN para que os mesmos enviassem essas informações através do barramento *dmesg* do Linux. Esses módulos foram instrumentados de modo que mensagens personalizadas relativas ao controlador sejam enviadas ao barramento. Essas mensagens então são capturadas pela aplicação do controlador, a qual realiza sua função de controle.

Além do envio de mensagens com as informações necessárias ao controlador, foram realizadas alterações no módulo L2CAP para remover o controle de créditos padrão do Kernel. No modelo padrão do Kernel, o controle de fluxo de créditos é realizado de maneira simples, onde a cada $20$ créditos consumidos por um cliente, o dispositivo mestre envia $20$ créditos adicionais àquele. Como o objetivo do controlador adaptativo apresentado nesse trabalho é sobrepor esse controle simples, todos os mecanismos de controle do Kernel para BLE foram desativados.

Para a identificação dos clientes foram utilizados o endereço IPv6 alocado ao mesmo em conjunto com seu endereço de identificação de hardware Bluetooth (endereço MAC). Essas informações são extraídas do módulo L2CAP (endereço MAC) e do módulo 6LoWPAN (endereço IPv6).

Para satisfazer o quarto requisito básico foi criado um mecanismo de envio de comandos de controle entre o controlador adaptativo e o módulo L2CAP. Como o controlador adaptativo reside em nível de aplicação e o módulo L2CAP em nível de Kernel, é necessário utilizar um mecanismo que permita a interação entre esses dois níveis de maneira simplificada. Para esse propósito foi utilizado o *debugfs*, o qual é um sistema de arquivo carregado em memória RAM utilizado inicialmente para propósitos de depuração de sistemas do Kernel. Nesse sentido, foi montado um sistema de arquivos simples para o compartilhamento de comandos entre o controlador adaptativo e o módulo L2CAP. Em um momento posterior, o controlador adaptativo deve ser integrado ao Kernel diretamente.

Com esses dois mecanismos de entrada e saída de dados implementados, o controlador adaptativo foi desenvolvido em nível de aplicação, o qual é executado como uma aplicação do tipo *daemon*. Essa aplicação, com isso, substitui o controlador de créditos do Kernel do Linux, podendo executar os algoritmos apresentados no Capítulo 5 com maior flexibilidade. A aplicação do controlador adaptativo foi desenvolvida de modo a permitir a alteração dos modelos de controle e seus algoritmos de maneira simplificada.

## 6.2 Metodologia de Avaliação e Configuração do Controlador

Após a implementação básica do controlador adaptativo para Gateways BLE, foi necessário realizar a avaliação individual de cada modelo de controle proposto. Para os todos os experimentos, foi definido um processo de testes único com o intuito de padronizar o processo de avaliação. Para cada experimento os passos descritos a seguir foram executados:

1. *Definição dos objetivos para cada experimento:* Cada experimento deve ter um objetivo claro, de modo que seja possível realizar observações e avaliações sobre seus resultados.

2. *Instrumentação do processo experimental nos clientes:* Os clientes devem ser instrumentados de modo que seja possível realizar a repetição dos cenários entre experimentos.

3. *Avaliação comportamental via inspeção gráfica entre experimentos:* A cada experimento, os resultados devem ser checados para verificar se o comportamento está condizente com o esperado.

4. *Discussão dos resultados relativos ao objetivo:* Além da observação dos resultados, os dados devem ser discutidos e avaliados estatisticamente em relação aos parâmetros definidos para dispositivos listados no objetivo.

Para todos os experimentos a seguinte configuração de rede foi considerada:

- Um dispositivo mestre com Controlador Bluetooth USB Broadcom BCM20702.
- No mínimo 3 dispositivos clientes com Controladores Bluetooth USB CSR8510 A10.

Para viabilizar o controle do processo experimental, todos os controladores Bluetooth USB clientes foram executados na mesma máquina Linux hospedeira. Dessa maneira foi possível controlar o início da transmissão de dados entre os clientes.

Para a geração de tráfego entre os clientes e o dispositivo mestre foi utilizada a ferramenta de rede *ping6*, a qual faz uso do *Internet Control Message Protocol - ICMP*. O *ping6* foi escolhido para que o mesmo tráfego de rede no sentido *uplink* fosse trafegado no sentido *downlink*. Desse modo é possível realizar uma avaliação homogênea do canal de transmissão. Adicionalmente, a ferramenta foi configurada para enviar pacotes utilizando o MTU máximo configurado para o canal BLE, e também foi configurada para sempre enviar um novo comando após receber a resposta do anterior. O propósito dessa configuração é permitir que o canal entre o cliente e o mestre sempre esteja utilizando o máximo da banda de transmissão disponível para o mesmo.

Por fim, para o uso do controlador adaptativo é necessário estabelecer parâmetros de configuração do Controlador Bluetooth. Para tanto, foram realizados experimentos para determinar a taxa máxima de transmissão entre um dispositivo cliente com controlador CSR e um mestre utilizando o controlador Broadcom. Os resultados são apresentados no gráfico da Figura 6.2 em relação ao número de créditos trocados por interação.

Observa-se que o controlador Broadcom tem uma taxa máxima de aproximadamente $2,6 KBytes/s$. Entretanto, o controlador apresentou instabilidade quando seu ciclo de funcionamento é mantido próximo dessa taxa máxima, por exemplo, a conexão entre dois dispositivos era abortada após alguns minutos de uso. Com isso, para os experimentos realizados

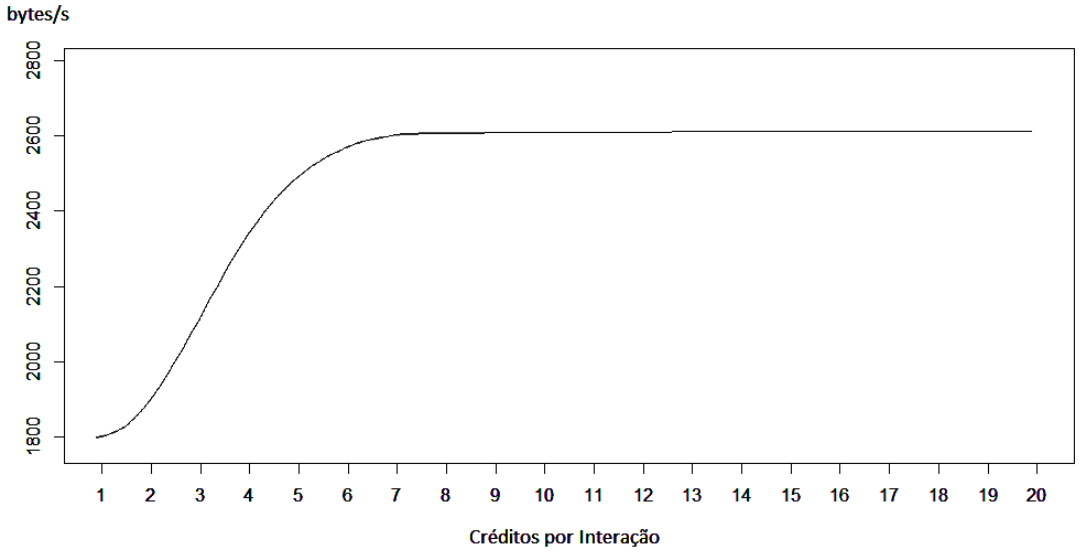


Figura 6.2: Crescimento da taxa de transmissão em relação ao aumento do número de créditos por interação para o chipset Broadcom BCM20702.

nesse trabalho, a taxa de $2,0KBytes/s$, ou aproximadamente $80\%$ da taxa máxima, foi escolhida como o taxa de transmissão máxima do controlador adaptativo para o controlador Broadcom.

Em relação aos créditos, utilizando o gráfico da Figura 6.2 como referência, o número de créditos padrão que devem ser trocados por interação estar entre o número 2 e 3 para a taxa máxima de $2,0KBytes/s$. Para os experimentos realizados nesse trabalho foi escolhido o limiar inferior de 2 créditos trocados por interação.

Em relação aos aspectos ambientais e físicos para a execução dos experimentos, os clientes e o dispositivo mestre foram dispostos a uma distância máxima de $50cm$ entre si. O ambiente de testes foi montado em uma área com objetivo de ter uma reduzida interferência entre dispositivos. Foram aferidas apenas duas redes IEEE 802.11 (Wi-Fi) com cobertura na mesma área, portanto, reduzindo a possibilidade de interferência com a rede BLE.

## 6.3 Avaliação do Controlador Simples

Esta seção apresenta detalhes sobre a avaliação experimental do Controlador Simples descrito na Seção 5.2.2 do Capítulo 5. Como descrito anteriormente, os parâmetros de

configuração utilizados para o controlador BLE foram:

- A taxa máxima de dados que o dispositivo controlador BLE suporta, $maxTx = 2000Bytes/s$.
- O número mínimo de créditos que devem ser enviados a um cliente por interação, $xCrs = 2$.

## Objetivos do Experimento

O principal objetivo desse experimento é avaliar o comportamento do Controlador Simples em relação a dois aspectos:

- Manutenção da taxa de transmissão máxima do Gateway BLE dentro do limiar com valor de $maxTx$.
- Divisão igualitária da taxa de transmissão $maxTx$ entre todos os dispositivos conectados.

## Processo Experimental

O procedimento definido para avaliar o objetivo do experimento consiste em:

1. Realizar a conexão individual de cada dispositivo cliente em diferentes instantes de tempo.
2. A cada conexão avaliar se a taxa total de transmissão de dados de cada cliente é reduzida e dividida igualmente.
3. Sempre avaliar se a taxa de transmissão $maxTx$ não é ultrapassada.

O procedimento foi repetido diversas vezes com conexões individuais em diferentes instantes de tempo.

## Avaliação Comportamental e Discussão dos Resultados

Em um total de 20 execuções do experimento, em todos os casos os objetivos foram alcançados. Como demonstração dos resultados, os gráficos apresentados na Figura 6.3 apresentam os resultados de duas repetições do experimento. Nessas duas repetições, os dispositivos conectam-se em instantes diferentes, e mesmo assim, ao final do experimento onde os três clientes estão conectados, o Controlador Simples fez o controle da taxa de transmissão individual de cada cliente.

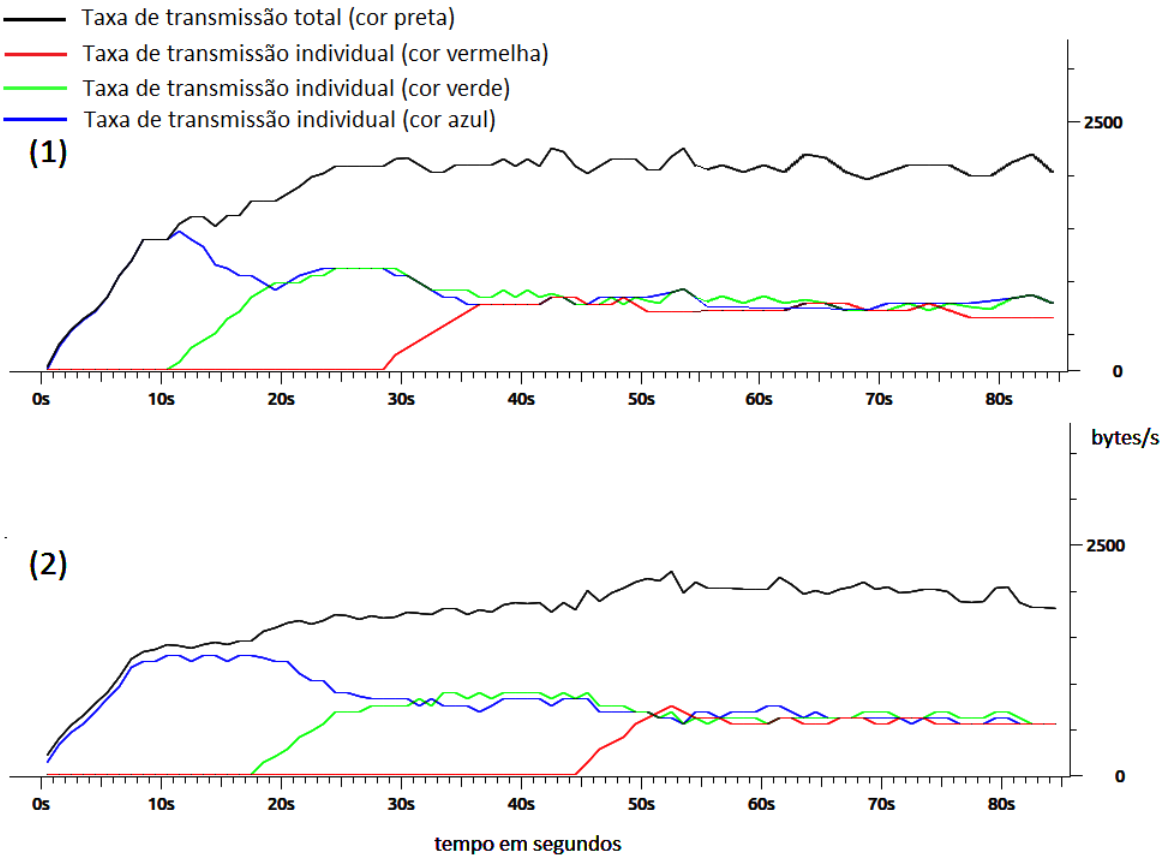


Figura 6.3: Gráficos representando com resultados do funcionamento do Controlador Simples.

No gráfico da Figura 6.3(1) os três dispositivos conectam-se ao Gateway BLE nos 30 primeiros segundos. No gráfico da Figura 6.3(2) as conexões são distribuídas dentro dos 50 primeiros segundos. Em ambos os casos, quando os dois primeiros dispositivos se conectam, a taxa individual de cada um fica em torno de $1KBytes/s$ e, após o terceiro dispositivo se conectar, a taxa individual de cada dispositivo é reduzida a aproximadamente $0,66KBytes/s$.

## 6.4 Avaliação do Controlador baseado em QoS

Esta seção apresenta detalhes sobre a avaliação experimental do Controlador Baseado em QoS descrito na Seção 5.2.3 do Capítulo 5. Como descrito anteriormente, os parâmetros de configuração utilizados para o controlador BLE foram:

- A taxa máxima de dados que o dispositivo controlador BLE suporta, $maxTx = 2000Bytes/s$.
- O número mínimo de créditos que devem ser enviados a um cliente por interação, $xCrs = 2$.

### Objetivos do Experimento

O principal objetivo desse experimento é avaliar o comportamento do Controlador Baseado em QoS em relação aos seguintes aspectos:

- Manutenção da taxa de transmissão máxima do Gateway BLE dentro do limiar com valor de $maxTx$.
- Divisão da taxa de transmissão $maxTx$ entre os dispositivos conectados proporcionalmente aos parâmetros de QoS de cada dispositivo, quando estes estão presentes.

### Processo Experimental

O procedimento definido para avaliar o objetivo do experimento consiste em:

1. Realizar a conexão individual de cada dispositivo cliente em diferentes instantes de tempo.
2. A cada conexão avaliar se a taxa total de transmissão de dados de cada cliente é ajustada proporcionalmente a sua taxa de transmissão definida no parâmetro de QoS, quando presente.
3. Sempre avaliar se a taxa de transmissão $maxTx$ não é ultrapassada.

O procedimento foi repetido diversas vezes com dispositivos conectando-se em diferentes instantes de tempo.

## Avaliação Comportamental e Discussão dos Resultados

Aproximadamente 20 execuções experimentais foram realizadas para avaliar esse controlador. Um primeiro conjunto de experimentos foram realizados para avaliar a prioridade de um dispositivo com parâmetros de QoS em relação aos outros. O gráfico da Figura 6.4 apresenta os resultados de um desses experimentos. Nesse experimento dois dispositivos estão conectados com o Gateway BLE compartilhando o meio igualmente. No ponto na Figura 6.4(a) o dispositivo com prioridade de QoS representado pela linha de cor $azul$ inicia sua conexão. Esse dispositivo tem como parâmetro de QoS utilizar $1KBytes/s$.

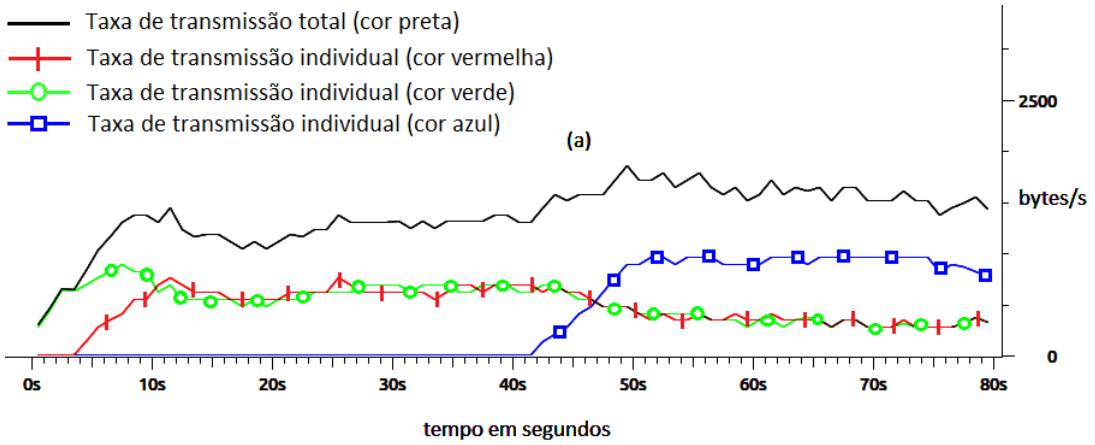


Figura 6.4: Gráfico com resultado de um experimento com o Controlador baseado em QoS.

É visível no gráfico da Figura 6.4 que a partir da conexão do dispositivo com parâmetros de QoS no ponto $(a)$, os outros dois dispositivos tem sua taxa de transmissão reduzida igualmente, como esperado dado o comportamento programado do controlador.

Em outro tipo de experimento, a ordem de conexão foi invertida, e o dispositivo com parâmetro de QoS foi conectado antes dos outros. O gráfico da Figura 6.5 apresenta os resultados de um desses experimentos.

No experimento da Figura 6.5, o dispositivo com parâmetros de QoS representado pela curva com cor $azul$ é o primeiro a se conectar. Após um segundo dispositivo se conectar, a taxa de transmissão permita do dispositivo com QoS é reduzida ao seu limiar desejado, o qual é de $1KBytes/s$. O segundo dispositivo conectado, representado pela curva de cor $vermelha$, tem sua taxa de transmissão limitada ao restante da taxa $maxTx$ disponível, a qual, coincidentemente, é de $1KBytes/s$. No instante representado pelo ponto da Figura

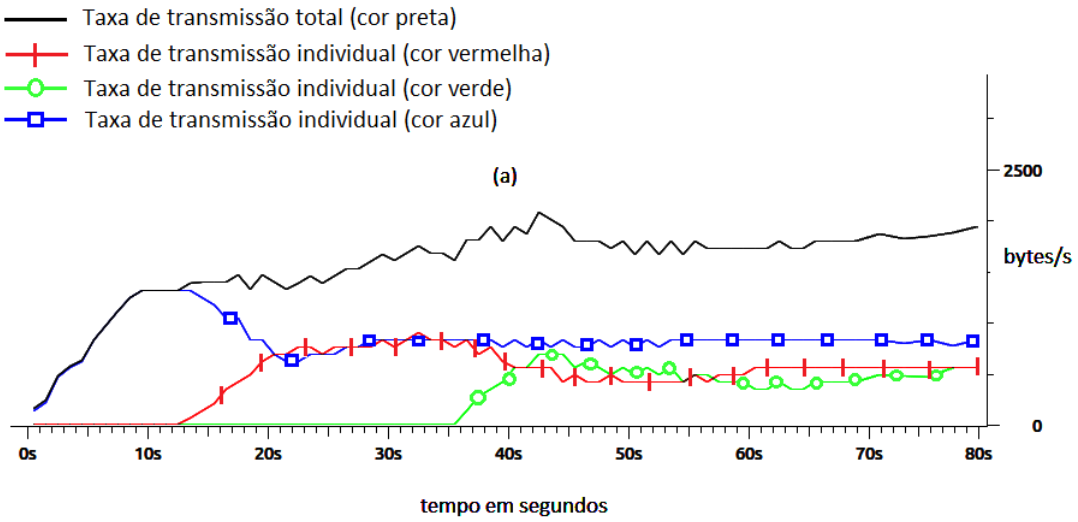


Figura 6.5: Gráfico com resultado de um segundo experimento com o Controlador baseado em QoS.

6.5(a), um terceiro dispositivo se conecta, o qual é representado pela curva de cor $verde$. O terceiro dispositivo também tem parâmetros de QoS definidos. Entretanto, pela definição do algoritmo do controlador apresentado na Seção 5.2.3, o mesmo define uma taxa de transmissão mínima que deve estar sempre disponível para novas conexões $minFreeTx$, a qual nos experimentos foi definida em $1KBytes/s$.

Como o primeiro dispositivo com QoS já está utilizando metade da taxa de transmissão, não é possível garantir a taxa de transmissão do novo dispositivo para seus parâmetros de QoS. Com isso, o mesmo foi tratado como um dispositivo sem parâmetros de QoS, e teve sua taxa de transmissão reduzida pelo Gateway BLE igualmente com o outro dispositivo sem parâmetros de QoS.

## 6.5 Avaliação do Controlador baseado em QoS com Prioridade

Esta seção apresenta detalhes sobre a avaliação experimental do Controlador baseado em QoS com definição de Prioridade descrito na Seção 5.2.4 do Capítulo 5. Como descrito anteriormente, os parâmetros de configuração utilizados para o controlador BLE foram:

- A taxa máxima de dados que o dispositivo controlador BLE suporta, $maxTx$ =

$2000 Bytes/s$.

- O número mínimo de créditos que devem ser enviados a um cliente por interação, $xCrs = 2$.

## Objetivos do Experimento

O principal objetivo desse experimento é avaliar o comportamento do Controlador Baseado em QoS com Prioridade em relação aos seguintes aspectos:

- Manutenção da taxa de transmissão máxima do Gateway BLE dentro do limiar com valor de $maxTx$.
- Divisão da taxa de transmissão $maxTx$ entre os dispositivos conectados proporcionalmente aos parâmetros de QoS de cada dispositivo, quando estes estão presentes.
- Aplicação da regra de prioridade entre os dispositivos, quando esta regra está presente.

## Processo Experimental

O procedimento definido para avaliar o experimento consiste nos seguintes passos:

1. Realizar a conexão individual de cada dispositivo cliente em diferentes instantes de tempo.
2. A cada conexão avaliar se a taxa total de transmissão de dados de cada cliente é ajustada proporcionalmente a sua taxa de transmissão definida no parâmetro de QoS, quando presente.
3. Avaliar se a regra de prioridade esperada é obedecida após a conexão de um novo dispositivo.
4. Sempre avaliar se a taxa de transmissão $maxTx$ não é ultrapassada.

O procedimento foi repetido diversas vezes com dispositivos conectando-se em diferentes instantes de tempo.

## Avaliação Comportamental e Discussão dos Resultados

Em torno de 20 execuções diferentes foram realizadas para avaliar esse controlador. Em princípio, esse controlador se assemelha muito com o anterior, diferenciando-se apenas da aplicação de prioridade entre dispositivos. Portanto, essa diferença foi explorada durante os experimentos.

Para explorar a funcionalidade de prioridade, os experimentos foram direcionados a sempre conectar um dispositivo com prioridade posteriormente a outro com prioridade menor. Os gráficos da Figura 6.6 apresentam duas execuções semelhantes utilizando o controlador com QoS e Prioridade. Nesses experimentos, um primeiro dispositivo com prioridade menor, mas com parâmetros de QoS se conecta a rede, o qual é representado pela curva de cor $verde$. Na configuração do controlador, a taxa de transmissão mínima que deve estar sempre disponível para novas conexões $minFreeTx$ foi definida em $1KBytes/s$.

Após alguns segundos, um dispositivo com maior prioridade, mas com os mesmos parâmetros de QoS se conecta a rede. Nesse instante, realizando o cálculo da taxa de transmissão livre $minFreeTx$, essa fica menor do que o limiar estabelecido pelo controlador. Com isso, é facilmente observado nos gráficos que o dispositivo com prioridade menor deixa de ter seus parâmetros de QoS garantidos, e passa a ser tratado como um dispositivo sem parâmetros de QoS.

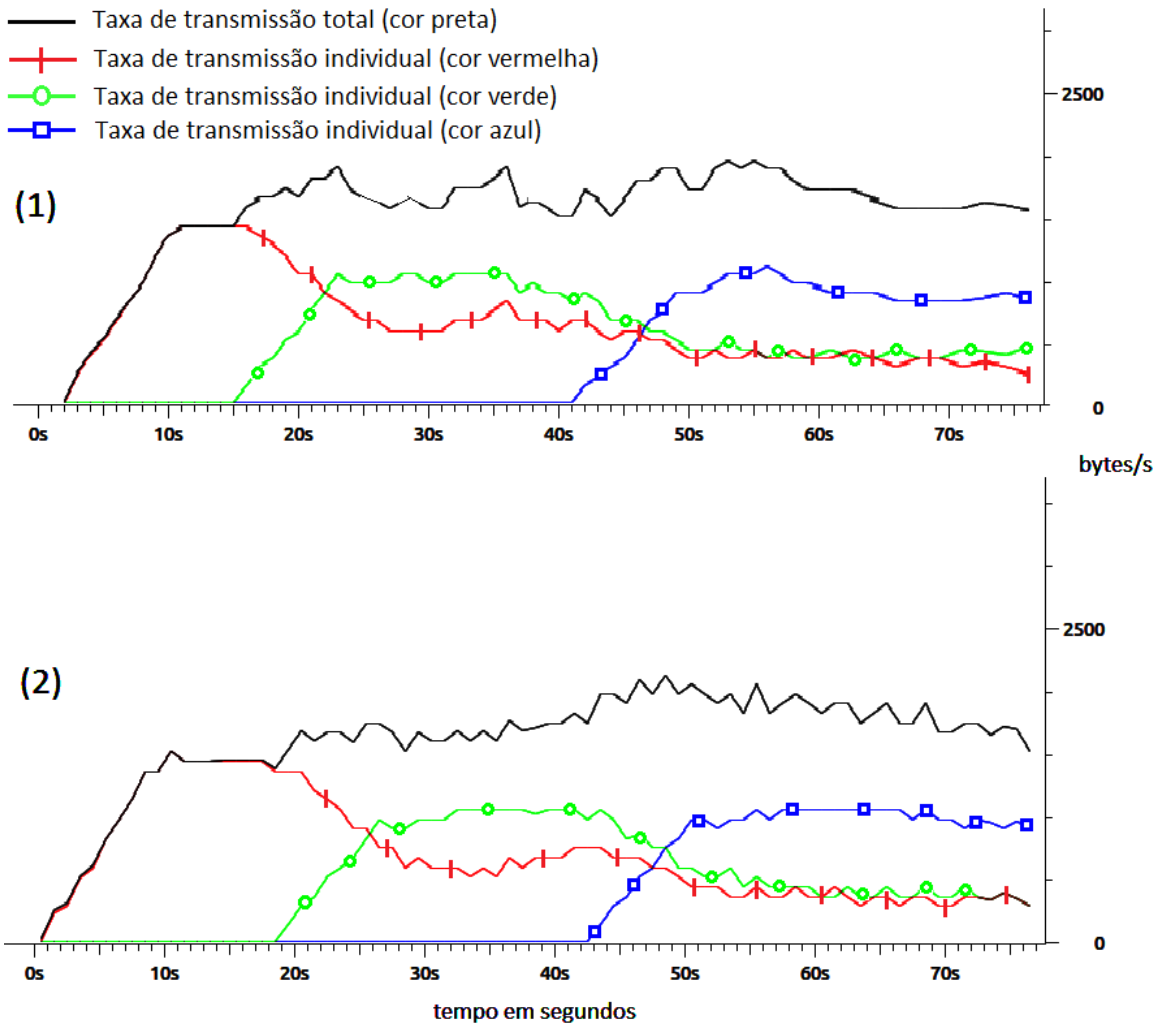


Figura 6.6: Gráficos com resultados do funcionamento do Controlador baseado em QoS com Prioridade.

Esse comportamento, como descrito na Seção 5.2.4 do Capítulo 5, se dá com o intuito de garantir que os parâmetros de QoS do dispositivo com maior prioridade sejam fornecidos, e para garantir que os demais dispositivos tenham um mínimo de taxa de transmissão disponível para manter suas conexões. Por exemplo, caso a taxa de transmissão mínima disponível $minFreeTx$ fosse zero, dois dispositivos com prioridade podiam manter seus parâmetros de QoS, caso a soma de suas taxas de transmissão desejadas não ultrapassem o valor máximo de $maxTx$.

## 6.6 Avaliação do Controlador baseado em QoS com Prioridade Temporal

Esta seção apresenta detalhes sobre a avaliação experimental do Controlador baseado em QoS com Prioridade Temporal descrito na Seção 5.2.5 do Capítulo 5. Como descrito anteriormente, os parâmetros de configuração utilizados para o controlador BLE foram:

- A taxa máxima de dados que o dispositivo controlador BLE suporta, $maxTx = 2000Bytes/s$.
- O número mínimo de créditos que devem ser enviados a um cliente por interação, $xCrs = 2$.

### Objetivos do Experimento

O principal objetivo desse experimento é avaliar o comportamento do Controlador Baseado em QoS com Prioridade Temporal em relação aos seguintes aspectos:

- Manutenção da taxa de transmissão máxima do Gateway BLE dentro do limiar com valor de $maxTx$.
- Divisão da taxa de transmissão $maxTx$ entre os dispositivos conectados proporcionalmente aos parâmetros de QoS de cada dispositivo, quando estes estão presentes.
- Aplicação da regra de prioridade entre os dispositivos de maneira temporizada, quando esta regra está presente.

### Processo Experimental

O procedimento definido para avaliar o experimento consiste nos seguintes passos:

1. Realizar a conexão individual de cada dispositivo cliente em diferentes instantes de tempo.
2. A cada conexão avaliar se a taxa total de transmissão de dados de cada cliente é ajustada proporcionalmente a sua taxa de transmissão definida no parâmetro de QoS, quando presente.

3. Avaliar se a regra de prioridade esperada é obedecida após a conexão de um novo dispositivo.

4. Avaliar se a taxa de transmissão de dados de um cliente com prioridade temporal é mantida apenas durante seu período de prioridade.

5. Sempre avaliar se a taxa de transmissão $maxTx$ não é ultrapassada.

O procedimento foi repetido diversas vezes com dispositivos conectando-se em diferentes instantes de tempo.

## Avaliação Comportamental e Discussão dos Resultados

Em torno de 20 experimentos diferentes foram realizados com o controlador com prioridade temporal. Nesses experimentos, assim como nos dois anteriores, as informações de prioridade, QoS e período de prioridade são informadas por aplicações externas através de uma interface do controlador. Por sua característica temporal, uma funcionalidade importante nesse controlador é deixada para aplicações externas: o reinício de prioridade de um dispositivo. Esse reinício consiste em informar o controlador adaptativo que a prioridade temporal de um dispositivo pode ser considerada novamente. Por exemplo, após atribuir prioridade a um dispositivo por $T$ segundos, esse dispositivo perde sua prioridade. Com o reinício, uma aplicação externa pode informar o controlador que aquele dispositivo deve ter prioridade de $T$ segundos novamente.

O gráfico na Figura 6.7 apresenta os resultados de um experimento com o controlador temporal. No ponto $(a)$ um primeiro dispositivo com parâmetros de QoS se conecta a rede. No instante no ponto $(b)$ um dispositivo com maior prioridade se conecta. Entretanto, esse dispositivo, representado pela curva de cor $azul$ tem uma prioridade temporal, e no instante do ponto (c) sua prioridade é encerrada e sua taxa de transmissão reduzida. Nesse instante, o dispositivo com parâmetro de QoS que já estava conectado volta a ter prioridade, e sua taxa de transmissão é aumentada para garantir seus parâmetros de QoS. Com isso, esse experimento exemplifica o comportamento de um dispositivo com prioridade temporal.

Para demonstrar a funcionalidade de reinício de prioridade, outro experimento foi realizado. O gráfico da Figura 6.8 apresenta os resultados desse experimento. De maneira

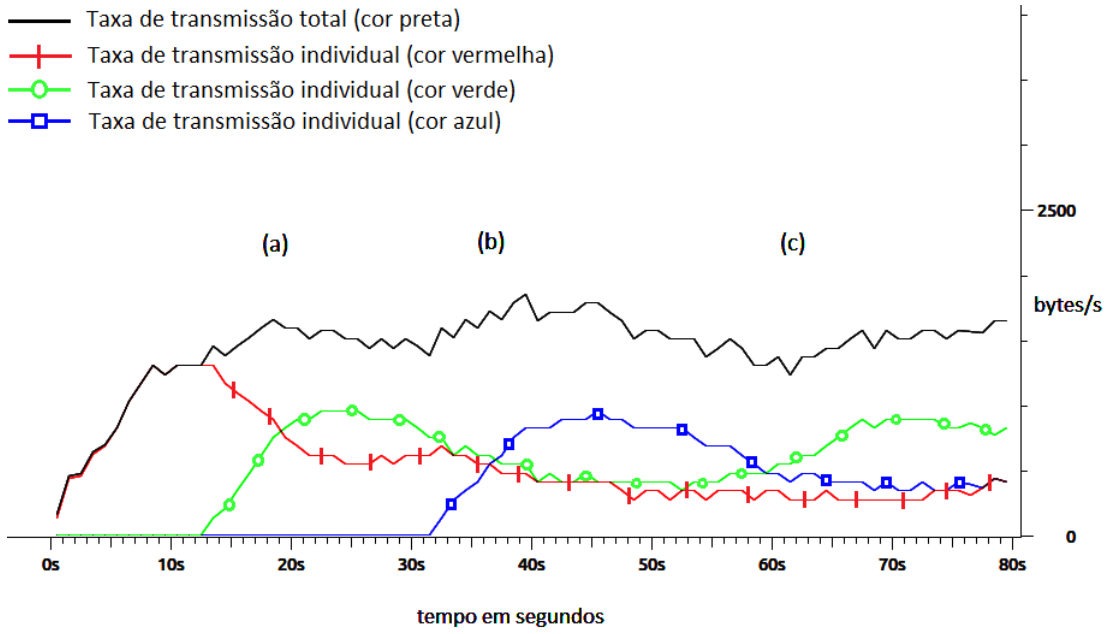


Figura 6.7: Gráfico com resultado do funcionamento de uma interação do Controlador baseado em QoS com Prioridade Temporal.

semelhante ao experimento descrito anteriormente, o dispositivo com prioridade temporal se conecta no instante representado pelo ponto $(b)$ e tem sua prioridade mantida até o ponto no instante $(c)$, onde a partir de então o dispositivo com parâmetros de QoS conectado no ponto $(a)$ volta a ter prioridade. A diferença em relação ao experimento anterior aparece no instante de tempo representado pelo ponto $(d)$, onde a prioridade do dispositivo representado pela curva de cor $azul$ é reiniciada. Nesse instante, o dispositivo $azul$ volta a ter prioridade até o instante de tempo representado pelo ponto $(e)$. Após esse instante de tempo, o dispositivo representado pela curva de cor $verde$ volta a ter prioridade como antes.

## 6.7 Discussão Geral dos Resultados

Como esperado, o controlador adaptativo conseguiu realizar o controle de fluxo entre os dispositivos clientes através de uma distribuição temporizada de créditos. Diversas situações foram testadas e avaliadas, mostrando que a proposta de utilizar o modelo de distribuição de créditos do BLE para garantir parâmetros de QoS de alguns dispositivos é viável.

Entretanto, algumas características devem ser observadas em relação a proposta desse controlador. Por exemplo, em alguns experimentos a taxa máxima de transmissão $maxTx$

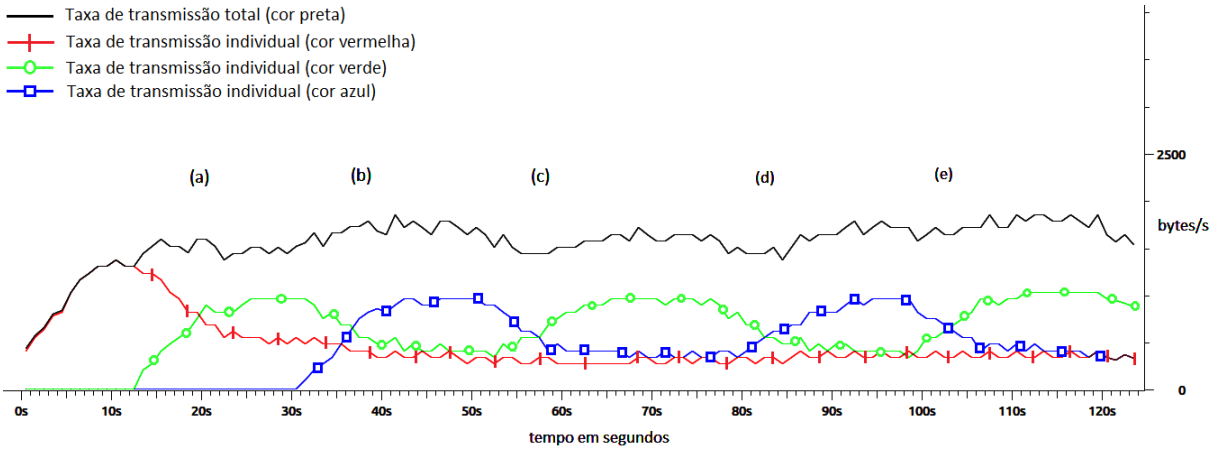


Figura 6.8: Gráfico com resultado do funcionamento de duas interações do Controlador baseado em QoS com Prioridade Temporal.

definida para o Gateway BLE não foi alcançada. Isso se deu ao fato de que mesmo que uma taxa de transmissão seja reservada para um cliente devido a seus parâmetros de QoS, não necessariamente significa que esse cliente irá utilizada em sua totalidade durante todo o período de tempo. Ou seja, podem existir situações as quais um Gateway BLE irá reservar uma taxa de transmissão $Tx$ a um cliente, mas esse cliente irá utilizar apenas $Tx/2$. Nessas situações, o controlador do Gateway BLE poderia realizar um ajuste nos parâmetros de QoS desse cliente, e liberar uma fração de sua taxa de transmissão para outros dispositivos.

Outro aspecto a ser considerado é o impacto dos pacotes de sinalização utilizados para a distribuição e créditos entre os clientes. Esses pacotes utilizam o mesmo canal de transmissão de dados e, portanto, oferecem um impacto na taxa de transmissão final para as aplicações IPv6 que estão utilizando o Gateway BLE.

Por fim, outra característica a ser apontada é a definição da taxa máxima de transmissão $maxTx$ do Gateway BLE. Apesar de o Controlador Bluetooth utilizado suportar uma taxa de transmissão ponto a ponto maior do que a definida nos experimentos, um valor reduzido foi configurado no controlador adaptativo por questões de instabilidade. Portanto, considerando um cenário de Internet das Coisas, onde diferentes dispositivos podem estar conectados ao mesmo Gateway simultaneamente, os fabricantes de *chipsets* BLE devem considerar cenários mais complexos durante o projeto de seus controladores de rádio.

## 6.8 Considerações Finais do Capítulo

Neste capítulo foram apresentados detalhes sobre o desenvolvimento do controlador adaptativo descrito no Capítulo 5. Além desses detalhes de implementação, foi descrita a metodologia de avaliação utilizada, a qual foi utilizada para avaliar individualmente cada modelo de configuração do controlador.

Após a definição da metodologia experimental, experimentos foram realizados para avaliar e validar cada modelo de controle aplicado ao controlador adaptativo. Em cada avaliação, experimentos foram realizados e seus resultados validados a partir de objetivos previamente definidos. Para cada experimento, uma avaliação e discussão de resultados foram realizadas e apresentadas.

Por fim, uma discussão geral dos resultados foi realizada, apontando possíveis encaminhamentos para trabalhos futuros, os quais serão descritos no Capítulo .

# Capítulo 7

# Controle de Fluxo de Dados aplicado a Gateways Pessoais para Saúde

A partir da arquitetura do Sistema de Monitoramento Remoto de Pacientes para a Internet das Coisas apresentada no Capítulo 4, nesse capítulo é apresentada uma avaliação da aplicação do Controlador Adaptativo do Capitulo 5 nesse sistema. Mais especificamente, foi desenvolvida uma aplicação de controle que utiliza informações de contexto fornecidas por aplicações externas. Essa aplicação realiza o envio de comandos de controle ao Controlador Adaptativo, portanto, realizando a adaptação da rede em relação aos dispositivos e ao contexto de saúde. Por fim, essa aplicação de controle foi implantada no *Smart-Gateway* de Saúde apresentado no Capítulo 4.

## 7.1 Arquitetura do *Smart-Gateway* para Saúde

O *Smart-Gateway* introduzido no Capítulo 4 realiza uma avaliação simples da rede, e como ação pode realizar o desligamento de conexões a depender do estado da rede PAN. Durante o seu desenvolvimento algumas características na avaliação de contexto para Sistemas de Monitoramento Remoto de Pacientes (SMRP) foram observadas. Dentre elas destacam-se as seguintes observações:

- *Ações em um processo de Monitoramento Remoto de Pacientes dependem do estado do paciente*. Por exemplo, as aplicações de saúde vão requisitar mais informações dos DPS de um paciente se o mesmo estiver com valores de pressão arterial alta;

- *Um processo de Monitoramento Remoto de Pacientes tem uma rotina*. Aplicações podem requisitar mais informações dos DPS de um paciente ao decorrer do tempo, de maneira orquestrada, criando um *Plano de Monitoramento*. Por exemplo, uma aplicação pode definir um *Plano de monitoramento* onde inicialmente é requisitada a pressão arterial, em seguida dez segundos de oximetria, e por fim mais dez segundos de ECG.

Considerando a arquitetura do *Smart-Gateway* apresentado em capítulos anteriores, essas duas observações requerem alterações no Monitor de QoS para Fluxos de Saúde (MQS). Em especial, dois módulos vão necessitar de alterações:

- o *Interpretador de Regras de Monitoramento (IRM)* precisa de um novo modelo de descrição de regras, o qual adicione suporte a *condições* e a variáveis de tempo para criação de *Planos de Monitoramento*;

- o *Avaliador de Fluxos de Saúde (AFS)* precisa de um novo formato de comandos de controle para o Controlador do Meio de Transporte (CMT). Esse formato deve adicionar variáveis de tempo para a orquestração de ações.

Esses novos comandos de controle são enviados ao Controlador Adaptativo do BLE desenvolvimento no Capítulo 5. Mais especificamente, foi utilizado o controlador baseado em QoS com prioridade temporal apresentado na Seção 5.2.5. Esse controlador permite que os requisitos apresentados anteriormente sejam alcançados, pois o mesmo permite priorizar dispositivos durante períodos específicos de tempo.

A seguir são apresentados detalhes sobre o novo modelo de descrição de regras de monitoramento, assim como os comandos de controles são enviados ao Controlador Adaptativo do Gateway BLE.

### 7.1.1 Novo Modelo de Descrição de Regras de Monitoramento

Com o objetivo de criar um *Plano de Monitoramento*, torna-se necessário estender a descrição de regras de monitoramento de modo a prover suporte a *condições* e variáveis de tempo.

Como descrito no Capítulo 5, regras de monitoramento utilizam como base a nomenclatura definida no padrão ISO/IEEE 11073:10101 [13]. Esse padrão define uma nomenclatura comum para diversas variáveis no contexto do ISO/IEEE 11073. No contexto do módulo de Interpretação de Regras de Monitoramento (IRM), apenas será utilizada a nomenclatura relativa a medidas (como unidades e dimensões) e de tipos de dispositivos (oxímetro de pulso, perfil de ECG, entre outros).

No modelo de IRM anterior, regras de monitoramento eram utilizadas apenas para indicar se um conjunto de DPS estava em um processo de MRP ou não. Portanto, *condições* simples eram avaliadas por operadores lógicos a cada novo evento, de modo a avaliar se um processo de MRP foi iniciado ou não. Alguns exemplos de regras são apresentados a seguir.

Código Fonte 7.1: Exemplos de Regras Simples de Identificação de Monitoramento

```
rule01:
when {MDC_DEV_SUB_SPEC_PROFILE_ECG starts AND last MDC_TEMP_BODY > 39}
then {(MDC_DEV_SUB_SPEC_PROFILE_ECG, MDC_DEV_SPEC_PROFILE_TEMP) is
    monitoring}

rule02:
when {(MDC_SYS_ID = 0011223344) starts}
then {(MDC_SYS_ID = 0011223344) is monitoring}
```

Em *rule01*, toda vez que um dispositivo com perfil de ECG inicia a transmissão e o último valor de temperatura corporal for maior que o inteiro 39, significa que o *Smart-Gateway* entrou em um processo de MRP para os dispositivos descritos em *then*. A regra *rule02* é mais simples, e apenas indica que quando o dispositivo com identificação "0011223344" inicia uma transmissão, significa que o *Smart-Gateway* entrou em um processo de MRP para os dispositivos descritos em *then*. Para ambos os casos, o MQS vai utilizar as informações de registro de cada DPS conectado à PAN para estabelecer os requisitos de QoS necessários para cada um deles.

A partir de agora, entretanto, as regras de monitoramento irão descrever *Planos de Monitoramentos*, ou seja, além de checar se uma condição foi satisfeita, um *Plano de Monitoramento* descreve o que irá acontecer nos próximos instantes de tempo com um determinado grupo de dispositivos. Alguns exemplos de planos são apresentados a seguir.

Código Fonte 7.2: Exemplos de Planos de Identificação e Monitoramento

```
plan01:
when {MDC_TEMP_BODY > 39}
then {MDC_DEV_SUB_SPEC_PROFILE_ECG starts after T1 seconds AND
   MDC_DEV_SPEC_PROFILE_TEMP starts after T2 seconds during Tx seconds}

plan02:
when {(MDC_SYS_ID = 0011223344) starts}
then {(MDC_SYS_ID = 4455667788) starts after T1 seconds}

plan03:
when {(MDC_HR > 110)}
then {MDC_DEV_SUB_SPEC_PROFILE_ECG starts after T1 seconds}
```

Com isso, o que os *Planos de Monitoramento* apresentam são previsões de como será o comportamento de certos dispositivos durante o processo de MRP. Por exemplo, o plano *plan01* descreve que quando um evento de temperatura for identificado com o valor maior que o inteiro 39, um dispositivo com o perfil MDC_DEV_SUB_SPEC_PROFILE_ECG irá iniciar um fluxo de transmissão após T1 segundos, e que outro dispositivo com um perfil MDC_DEV_SPEC_PROFILE_TEMP irá iniciar um fluxo de transmissão após T2 segundos durante Tx segundos. Com essas informações, em conjunto com as informações de registro de cada dispositivo, o módulo de Avaliador de Fluxos de Saúde cria um *Plano de Monitoramento* detalhado através de Comandos de Controles, os quais são enviados ao Controlador do Meio de Transporte (CMT).

### 7.1.2 Novo Modelo de Descrição de Comando de Controle

Dado as alterações propostas, um Comando de Controle deve adicionar o instante de tempo em que o DPS deve iniciar uma transmissão, e uma previsão de tempo ao qual essa transmissão deve ser mantida. Portanto, considera-se um Comando de Controle como o conjunto das seguintes informações:

**Definição 5 (Novo Comando de Controle)** .

$$ComandoDeControle = \{DeviceId, Priority, Tx_{QoS}, P_0, P_{MAX}\}$$

*Onde:*

$DeviceId$ *é a identificação do DPS.*

$Priority$ *é um inteiro com uma indicação de prioridade.*

$Tx_{QoS}$ *a taxa ideal de transmissão para alcançar os requisitos de QoS necessários.*

$P_0$ *é o período de tempo após o qual esse DPS vai iniciar a transmissão*

$P_{MAX}$ *é uma estimativa de quanto tempo esse DPS deve manter sua taxa de transmissão.*

## 7.2 Desenvolvimento e Integração do Avaliador de Fluxo de Saúde

Como discutido em capítulos anteriores, a depender do contexto relativo ao processo de Monitoramento Remoto de Pacientes, um conjunto específico de dispositivos requer que seus parâmetros de QoS sejam garantidos a nível de rede. Portanto, o Controlador do Meio de Transporte (CMT) para rede PAN BLE precisa aplicar um novo modelo de distribuição de créditos para viabilizar o suporte aos parâmetros de QoS para alguns dispositivos por períodos de tempo determinados utilizando o controlador adaptativo do Capítulo 5.

Para definir quais dispositivos, requisitos, e o período de tempo ao qual os mesmos necessitarão de canais com garantia de QoS, o CMT faz uso dos Comandos de Controle recebidos do Avaliador de Fluxo de Saúde. Com isso, a partir da interpretação do Comando de Controle, o CMT irá realizar uma distribuição dinâmica e temporal de créditos com os dispositivos conectados à rede PAN BLE, de modo a suprir os requisitos de QoS para os DPS participantes do processo de monitoramento.

Seguindo o mesmo modelo apresentado no Capítulo 4, para o desenvolvimento do *Smart-Gateway* com Controle de Fluxo Adaptativo foram desenvolvidos novos módulos os quais foram integrados ao Controlador de Meio de Transporte. O novo modelo arquitetural é apresentado no diagrama da Figura 7.1 e são descritos a seguir:

- o módulo *Avaliador de Fluxo de Dados (PCAP)*. Esse módulo é responsável por escutar o fluxo de dados na rede IPv6, como representado na Figura 7.1(a). Observando filtros pré-definidos recebidos de outros módulos, a cada vez que uma regra de um filtro é obedecida, esse módulo sinaliza um evento para os módulos interessados.

- o módulo central *Avaliador de Fluxos de Saúde (AFS)*. Esse módulo é responsável por receber eventos sobre o fluxo de dados enviado pelo avaliador de fluxo de dados, como apresentado na Figura 7.1(b). Esse módulo tem interfaces para recebimento de regras de aplicações (Figura 7.1(c)), recebimento de eventos do Avaliador de Fluxo de Dados (Figura 7.1(b)), e envio de Comandos de Controle (Figura 7.1(d)). Esse módulo também integra as funções dos módulos de *Registro de Dispositivos (RD)*, e do módulo de *Interpretador de Regras de Monitoramento (IRM)*.

- o módulo Controlador do Meio de Transporte (CMT). Esse módulo tem acesso direto ao módulo Bluetooth da plataforma. Ele tem uma interface de entrada para recebimento dos Comandos de Controle (Figura 7.1(d)). Esse módulo realiza o controle de fluxo de dados através do Controlador Adaptativo.

O módulo *Avaliador de Fluxo de Dados*, por estar conectado diretamente com o módulo Avaliador de Fluxos de Saúde, acaba por interpretar apenas regras relativas a dados de saúde, portanto, fazendo o papel do módulo 11073Ex do Capítulo 4. Também, relativo a arquitetura do Capítulo 4, o módulo de regras realizava uma avaliação das mesmas para determinar se um processo de monitoramento estava em execução. Nesse novo modelo arquitetural de software, essa avaliação foi transferida para o módulo AFS, o qual também realiza a interpretação de regras.

Considerando a arquitetura de software proposta, foram definidas algumas interfaces entre os módulos, como ilustrado na Figura 7.1. A seguir são apresentadas as principais interfaces e seus requisitos:

- *Interface de Dados de Tráfego (Interface A1).* Essa interface é utilizada pelo Avaliador de Fluxo de Dados para enviar eventos ISO/IEEE 11073 trafegados pela rede IPv6. Essa interface é representada na Figura 7.1(b);

- *Interface de Regras de Monitoramento (Interface A2).* Essa interface é utilizada pelo AFS para recebimento de *Planos de Monitoramento* de aplicações, como apresentado na Figura 7.1(c);

- *Interface de Comandos de Controle (Interface A4).* Essa interface é utilizada pelo AFS para o envio de Comandos de Controle ao controlador, como apresentado na (Figura

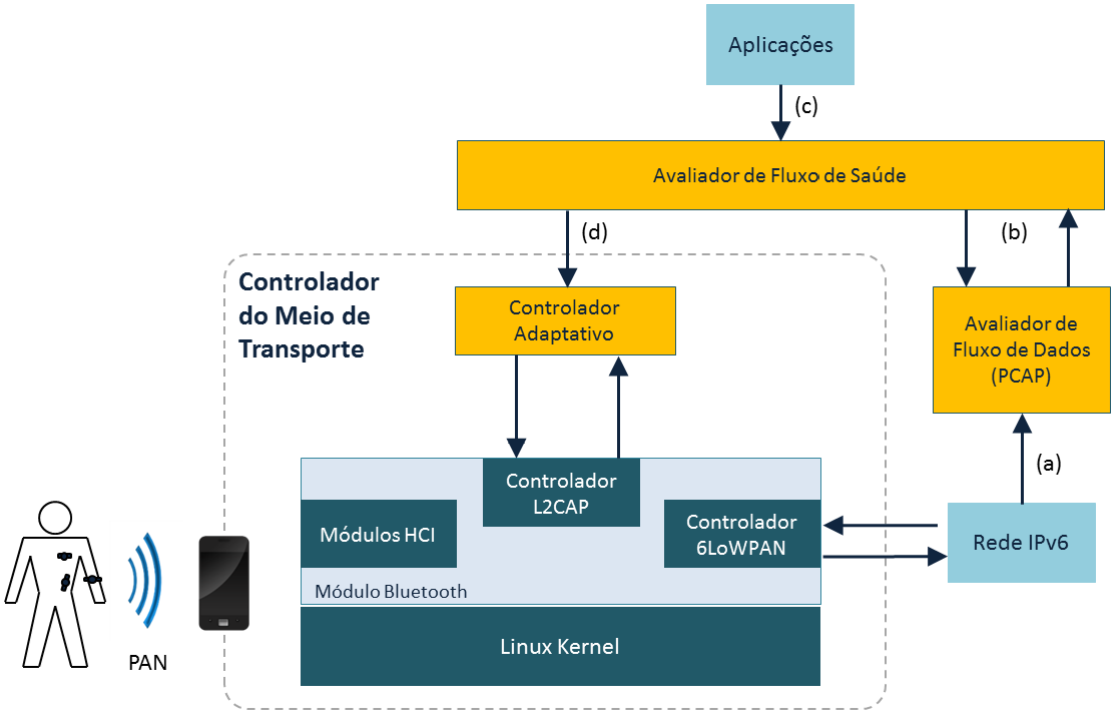


Figura 7.1: Diagrama com Modelo Arquitetural do Controlador de Fluxo com um Avaliador de Fluxo de Saúde.

7.1(d)).

Com essas definições, o funcionamento do *Smart-Gateway* pode ser descrito através da Figura 7.1:

1. Aplicações enviam regras relativas ao um processo de monitoramento ao Avaliador de Fluxo de Saúde em (c);

2. O Avaliador de Fluxo de Saúde interpreta essas regras, e envia filtros específicos ao Avaliador de Fluxo de Dados em (b);

3. O Avaliador de Fluxo de Dados inicia o processo de avaliação da rede IPv6 em (a);

4. Ao identificar um padrão de algum filtro recebido, o Avaliador de Fluxo de dados envia um evento ao Avaliador de Fluxo de Saúde em (b);

5. Caso o evento indique o início de um processo de monitoramento, o Avaliador de Fluxo de Saúde envia comandos de controle ao CMT em (d), assim, finalizando o processo.

### 7.2.1 Ferramentas e Arcabouços de Desenvolvimento

Para o desenvolvimento de um protótipo do *Smart-Gateway* para Saúde, além do controlador desenvolvido em capítulos anteriores, foram desenvolvidos dois novos módulos: O Avaliador de Fluxo de Dados e o Avaliador de Fluxos de Saúde.

Em paralelo, um novo modelo de comunicação para dados de saúde foi utilizado. Nesse novo modelo, os DPS ao invés de enviar pacotes ISO/IEEE 11073 diretamente aos serviços de saúde, enviam pacotes HL7 ORU (HL7 *Observation Response Unit*) [47]. Apesar de ser outro formato de pacote, a representação de dados de dispositivos de saúde ainda seguem o padrão semântico do ISO/IEEE 11073 [1], portanto, sendo compatível com a arquitetura apresentada no Capítulo 4. O HL7 ORU foi escolhido para esse protótipo dado a sua maior facilidade de interpretação em relação a APDUs ISO/IEEE 11073. Entretanto, todos os resultados aqui obtidos também se aplicam a APDUs ISO/IEEE 11073.

Código Fonte 7.3: Exemplo de Mensagem PCD-01 HL7 ORU

```
1
2 MSH|^~\&|Embedded^ACDE48234567ABCD^EUI-64||||20090713090030+0000||ORU^
      R01^ORU_R01|MSGID1234|P|2.6|||NE|AL|||||IHE PCD ORU-R01 2006^HL7
      ^2.16.840.1.113883.9.n.m^HL7\r
3 PID|||{USERID}^^^Imaginary Hospital^PI ||Doe^John^Joseph^^^^L^A|||M\r
4 OBR|1|AB12345^AcmeAHDInc^ACDE48234567ABCD^EUI-64|CD12345^AcmeAHDInc^
      ACDE48234567ABCD^EUI-64|182777000^monitoring of patient^SNOMED-CT|||{
      DATE}+0000\r
5 OBX|1||528388^MDC_DEV_SPEC_PROFILE_PULS_OXIM^MDC|1|||||||X|||{DATE
      }+0000||||1234567890AABBCCDDEEFF^EUI-64\r
6 OBX|2|NM|150456^MDC_PULS_OXIM_SAT_O2^MDC|1.0.0.1|80.5|262688^
      MDC_DIM_PERCENT^MDC||DEMO~ALINH|||R|||{DATE}+0000\r
7 OBX|3|CWE|67911^MDC_ATTR_MSMT_STAT^MDC|1.0.0.1.1|1^demo-data(5)||||||R\r
8 OBX|4|NA|150452^MDC_PULS_OXIM_PLETH^MDC
      |1.0.0.2|12^123^24^12^234^55^66^77^88^99|262656^MDC_DIM_DIMLESS^MDC
      |||||R|||{DATE}+0000
```

O Código Fonte 7.3 apresenta um pacote HL7 ORU enviado por um dispositivo do tipo oxímetro. Nesse pacote, consegue-se observar diversos campos de identificação. Em especial, observando a *linha 6*, é possível identificar o dado do tipo *MDC_PULS_OXIM_SAT_O2*,

[1] https://rtmms.nist.gov/

o qual representa um dado de saturação de oxigênio no sangue. Na mesma linha, mais adiante, observa-se o valor $80.5$ e a indicação *MDC_DIM_PERCENT*, ou seja, essa linha indica o envio de uma observação com o resultado de $80.5\%$ de saturação sanguínea. Todas essas definições estão presentes nas especificações do HL7 [75] e nas recomendações do IHE [47]. Também é importante ressaltar que os nomes com prefixo *MDC* seguem a semântica do ISO/IEEE 11073. Por fim, apesar de usar um novo modelo de pacote, o protocolo de transporte se manteve o CoAP, como discutido no Capítulo 4.

Para o Avaliador de Fluxo de Dados (AFD) foi desenvolvido um módulo de escuta baseado no pacote PCAP (*Packet Capture*) da linguagem de programação Python[2]. Com esse pacote é possível definir a interface de comunicação a ser escutada, nesse caso a interface BLE, e aplicar filtros de avaliação através de expressões regulares aos pacotes de dados. Com isso, o módulo AFD realiza a escuta especifica do conteúdo de pacotes IPv6 que trafegam na interface BLE.

Portanto, para o AFD, a *Interface A1* recebe sempre dois parâmetros: uma expressão regular do que deve ser filtrado e encontrado; e uma identificação de retorno quando a expressão regular for encontrada (*call-back*). A identificação serve para que o módulo que receba o evento de *call-back* mapeie qual expressão regular (regra) foi encontrada.

O Avaliador de Fluxos de Saúde (AFS) por sua vez foi desenvolvido como módulo integrador. Ele recebe regras de monitoramento de aplicações e cria filtros com expressões regulares simples, as quais são enviadas ao AFD para escuta da rede BLE. Ao receber o *call-back* do módulo AFD, o AFS faz o mapeamento de qual regra de monitoramento está em execução, e envia comandos de controle ao controlador adaptativo quando necessário.

## 7.3 Avaliação do *Smart-Gateway* com Controle de Fluxo na Rede PAN

Para avaliação do *Smart-Gateway* com controle de fluxo adaptativo experimentos foram realizados de modo à observar vários parâmetros de funcionamento dos DPS participantes de um processo monitoramento. O processo experimental utilizado foi o mesmo definido no

[2]https://pypi.python.org/pypi/pypcap

Capítulo 6.

Como exemplificação do funcionamento do controlador adaptativo em um contexto de monitoramento remoto de pacientes, os resultados de um experimento são discutidos a seguir.

## Objetivos do Experimento

O principal objetivo desse experimento é avaliar o comportamento do *Smart-Gateway* em um contexto de aplicações de saúde observando os seguintes aspectos:

- Manutenção da taxa de transmissão máxima do Gateway BLE dentro do limiar com valor de $maxTx$.
- Divisão da taxa de transmissão $maxTx$ entre os DPS conectados proporcionalmente aos parâmetros de QoS de cada dispositivo, quando estes estão presentes.
- Aplicação da regra de prioridade entre os DPS participantes do processo de monitoramento.
- Identificação de situações de monitoramento (contexto) a partir de regras pré-definidas e avaliação do fluxo de dados em uma rede BLE.
- Funcionamento de DPS em diferentes momentos de um processo de monitoramento.

## Processo Experimental

O procedimento definido para avaliar o experimento consiste nos seguintes passos:

1. Realizar a conexão individual de um dispositivo cliente que não consiste em um DPS.
2. Conectar um DPS que após $T$ segundos envia uma condição que dispara uma regra de um plano de monitoramento.
3. Avaliar se a regra de prioridade para os DPS participantes do plano de monitoramento é obedecida.
4. Instrumentar um DPS para esperar um comando de início (gatilho) para iniciar a transmissão de dados.

5. Sempre avaliar se a taxa de transmissão $maxTx$ não é ultrapassada.

6. Observar se o processo de monitoramento é encerrado.

O procedimento foi repetido diversas vezes com dispositivos conectando-se em diferentes instantes de tempo.

## Avaliação Comportamental e Discussão dos Resultados

O gráfico da Figura 7.2 apresenta os resultados do seguinte experimento:

1. um dispositivo de uso geral foi conectado ao *Smart-Gateway*, o qual é representado pelo gráfico de cor $vermelha$. Esse dispositivo, nos experimentos, envia comandos do tipo $ping6$ de maneira continua ao Gateway.

2. no instante do ponto (a), um DPS do tipo oxímetro conecta-se a rede e começa a enviar dados de maneira continua a um servidor de saúde na rede local. Nesse instante, nenhum processo de monitoramento está em execução e, portanto, os dois dispositivos conectados dividem os recurso de rede igualmente. Ressalta-se também que um terceiro dispositivo representado pela cor $azul$, encontra-se conectado, mas não envia dados;

3. no instante do ponto (b), o DPS oxímetro envia um dado que dispara uma regra de monitoramento. Nesse mesmo instante, o Avaliador de Fluxo de Saúde envia um comando de controle ao controlador, indicando que o dispositivo de cor $azul$ terá prioridade temporal durante $30s$;

4. em paralelo, uma aplicação de saúde envia um comando de requisição de dados ao DPS de cor $azul$, o qual inicia a transmissão de dados no ponto (b) e tem seus parâmetros de QoS garantidos.

5. o processo de monitoramento indicado pela regra disparada em (b) também indica que após $30s$ o DPS oxímetro terá uma prioridade temporal de outros $30s$. Portanto, no ponto (c), o DPS de cor $azul$ para de enviar dados, e o DPS de cor $verde$ ganha prioridade na transmissão de dados até o ponto (d).

6. a partir do ponto (d), apenas os demais dispositivos continuam conectados e transmitindo dados.

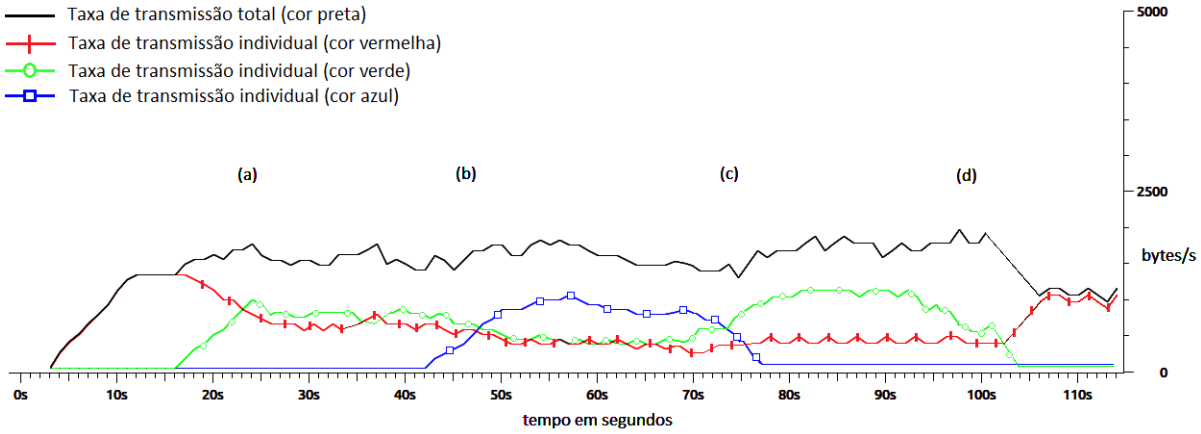


Figura 7.2: Gráfico com resultados experimentais da avaliação de um Smart-Gateway com aplicações de saúde.

O exemplo aqui discutido apresenta diversos detalhes sobre o funcionamento do *Smart-Gateway* em conjunto com as funcionalidades do controlador adaptativo BLE apresentado no Capítulo 5. Em especial, deve-se observar que o controle temporal do controlador adaptativo é essencial para aplicação de planos de monitoramento. Assim como, deve-se observar que o uso de um controle adaptativo no Gateway permite que outros dispositivos, como o dispositivo de cor $vermelha$ sempre se mantenham conectados a rede, mesmo quando processos de monitoramento remoto de pacientes estão em execução. Ou seja, o *Smart-Gateway* permite uma distribuição mais justa de recursos de rede entre todos os dispositivos, e garante os requisitos de QoS dos dispositivos que estão em um processo de monitoramento.

## 7.4 Considerações Finais do Capítulo

Nesse capítulo foram apresentados detalhes sobre a aplicação do controle adaptativo em redes BLE aplicado a um *Smart-Gateway* para a saúde. Foram detalhados quais os requisitos para a definição de um *Plano de Monitoramento*, quais as informações de contexto utilizadas, como elas são comparadas com regras, e como são gerados comandos para um *Plano de Monitoramento*.

No *Smart-Gateway* proposto, foram apresentados detalhes sobre como vai ser realizado o Controle de Fluxo de Dados em um Gateway pessoal, especificamente, um Gateway para redes BLE. A abordagem proposta faz uso do controlador adaptativo para redes BLE, o qual faz uso de um modelo de alocação dinâmica de créditos na camada de enlace. Portanto, a partir dessa abordagem, fica possível garantir requisitos de QoS para DPS participantes de um processo de monitoramento, a partir dos comandos de controle que descrevem o *Plano de Monitoramento* obtido através de informações de contexto fornecidas por aplicações e serviços.

Por fim, foram apresentados detalhes da implementação de um *Smart-Gateway* para saúde, dando ênfase as ferramentas utilizadas e como foram realizadas as interações entre os módulos. Também foram apresentados resultados de experimentos os quais exemplificam e validam o *Smart-Gateway* proposto.

# Capítulo 8

# Considerações Finais

Novas tecnologias voltadas para a Internet das Coisas (IoT) estão sendo criadas, as quais viabilizam a criação de novos serviços e aplicações em diversas áreas. Em paralelo, tecnologias de comunicação estão sendo estendidas com o objetivo de viabilizar a conexão da maior variedade de dispositivos com a Internet, como por exemplo, Dispositivos Pessoais de Saúde (DPS). Nesse sentido, tecnologias amplamente adotadas por dispositivos pessoais, como *smartphones*, estão criando extensões com foco no IoT. Esse é o caso do Bluetooth Low-Energy e o perfil *IP Service Profile* [7] o qual permite a transmissão de dados IPv6.

Com essas novas tecnologias, a evolução de DPS conectados e o uso de novos Serviços de Monitoramento Remoto de Pacientes (SMRP), viabiliza a criação de redes pessoais (PAN) onde DPS e sensores biométricos utilizam Gateways pessoais para compartilhar dados de saúde e sinais vitais com a Internet. Entretanto, essas mesmas redes são compartilhadas com outros dispositivos de fins diversos, como dispositivos de áudio e vídeo, *smart-glasses* e *smart-watches*, entre outros.

Considerando o cenário descrito, algumas características específicas de sistemas de monitoramento devem ser consideradas para o seu funcionamento confiável de uma rede PAN. Por exemplo, um sistema de monitoramento pode requisitar que alguns sensores enviem dados em conjunto para que seja possível realizar um diagnóstico. Entretanto, devido ao compartilhamento da rede com outros dispositivos, os sensores participantes desse processo de monitoramento podem não ter acesso aos requisitos de Qualidade de Serviço (QoS) necessários para a transmissão. Isso se deve ao fato de redes PAN, como redes BLE, não oferecerem diferenciação de canais, portanto, oferecendo apenas conexões do tipo *best-effort*.

O trabalho apresentado nesse documento se encaixa nesse contexto, onde DPS e sensores participantes de um processo de monitoramento de pacientes precisam ter a garantia de que requisitos de QoS sejam fornecidos para o compartilhamento de dados. Revisitando o cenário apresentado e os requisitos discutidos no decorrer desse trabalho, o Controlador de Fluxo Adaptativo para Gateways Bluetooth Low-Energy aplicado a Sistemas de Monitoramento Remoto de Pacientes oferece uma solução para:

- Aplicar um controle de fluxo de dados na camada de enlace do BLE de maneira dinâmica. Essa abordagem utiliza um método de alocação de créditos, onde os créditos são distribuídos dinamicamente de acordo com a prioridade e os parâmetros de QoS de um dispositivo.

- Definir dinamicamente a prioridade e os parâmetros de QoS de cada dispositivo a depender da função definida a nível de aplicação no dispositivo Gateway BLE.

- A partir do contexto de monitoramento do paciente que está utilizando o Gateway BLE, criar um *Plano de Monitoramento* para que o Gateway possa garantir os requisitos de QoS dos DPS e sensores participantes de um processo de monitoramento.

- Descrever regras de monitoramento de pacientes a partir de uma linguagem padronizada baseada em protocolos amplamente utilizados em Sistemas de Monitoramento Remoto de Pacientes.

- Evitar que o hardware Bluetooth do Gateway BLE ultrapasse seus limites de funcionamento e entre em um estado de conexão instável.

Com isso, a solução apresentada oferece uma abordagem multicamada, onde informações de contexto em nível de aplicação podem ser utilizadas para o controle de fluxo de dados em nível de enlace do BLE. Essa solução, portanto, consegue garantir que os requisitos de QoS dos DPS em um Gateway pessoal durante processos de monitoramento sejam alcançados, oferecendo ao usuário do serviço de monitoramento um sistema mais seguro e confiável.

Do ponto de vista do controlador de hardware BLE, a solução apresentada utiliza informações de limitações do hardware para realizar uma melhor distribuição de recursos

entre os dispositivos conectados. Essa distribuição de recursos permite tanto poupar os limites do Gateway BLE, quanto tentar garantir os requisitos de QoS necessários aos clientes conectados.

A partir de avaliações experimentais, foi demonstrada a viabilidade de se realizar um controle de fluxo adaptativo a partir do controle da distribuição de créditos na camada de enlace do BLE. Além disso, o controlador adaptativo foi aplicado a um contexto de Sistemas de Monitoramento Remoto de Pacientes, validando sua utilidade para esse tipo de sistemas, tendo como alvo um Sistema de Monitoramento Remoto de Paciente na Internet das Coisas. Por fim, durante as avaliações experimentais foram demonstradas limitações de adaptadores BLE, e como essas limitações podem ser controladas através do controlador adaptativo.

## 8.1 Perspectivas e Trabalhos Futuros

Em relação as perspectivas e possíveis trabalhos futuros derivados deste trabalho de pesquisa, destacam-se os seguintes itens:

- O controlador proposto faz uso de informações oferecidas por aplicações e pelos próprios dispositivos. Em um possível trabalho futuro, o controlador pode realizar uma avaliação instantânea das condições de rede, e realizar uma distribuição *just-in-time* dos créditos entre os dispositivos de rede a depender da demanda de cada um naquele instante.

- Para realizar uma avaliação mais rápida do controlador e seus algoritmos atuais (e futuros), pode ser desenvolvido um simulador de redes BLE com controle de fluxo baseado em créditos. Até a escrita desse trabalho, nenhum trabalho relativo a simuladores BLE com distribuição de créditos foi encontrado. Atualmente, em andamento, está sendo realizado um trabalho de modelagem através de redes de Petri Coloridas para o controle de fluxo baseado em créditos do BLE. A partir desse trabalho pretende-se verificar o funcionamento atual dos algoritmos propostos, assim como avaliar e desenvolver novos algoritmos.

- O modelo de controle fluxo adaptativo do BLE pode ser aplicado a outros domínios, criando a possibilidade de avalia-lo em diferentes situações. A partir de sua aplicação

em outros domínios, como multimídia, o controle adaptativo de fluxo pode ser validado para diferentes situações.

- O trabalho exploratório de avaliação e adaptação do controle de fluxo baseado em créditos do BLE pode ser aplicado em novos usos da tecnologia. O BLE, por padrão, não suporta redes do tipo *mesh*. Entretanto, trabalhos estão sendo desenvolvidos e soluções comerciais já estão disponíveis as quais permitem utilizar o BLE em redes *mesh*. Com isso, possibilidades aparecem, as quais podem permitir, através de um controle de fluxo baseado em créditos, realizar um controle de fluxo distribuído em redes *mesh* objetivando-se evitar congestionamento em nós intermediários.

- Implementar o controlador atual em uma plataforma móvel, e definir um modelo de acesso para que diferentes aplicações possam compartilhar as informações do Gateway BLE. Além disso, a partir do momento em que várias aplicações têm acesso e podem controlar o Gateway, regras de prioridade e segurança devem ser definidas em nível de plataforma móvel.

- Inserir o controlador adaptativo e o Gateway BLE a um cenário mais amplo da Internet das Coisas. Recentemente, novas arquiteturas e arcabouços de comunicação para o IoT estão sendo propostos, os quais também fazem uso do BLE como tecnologia de transporte, como por exemplo o *Open Connectivity Foundation*[1]. Com a adoção crescente do BLE e sua capacidade de transporte de dados IPv6, consequentemente esses arcabouços precisarão de propostas de controle inteligente no primeiro nível de conexão, a rede PAN.

- Evoluir e avaliar novas tecnologias e padrões de comunicação para a Internet das Coisas aplicada a saúde. Recentemente, novos protocolos para a transmissão de dados em saúde estão sendo propostos, como é o caso do FHIR[2]. Esses novos protocolos podem ser transportados por protocolos propostos nesse trabalho de pesquisa, como o CoAP. Entretanto, apesar de também seguirem o ISO/IEEE 11073 como referência na representação de dispositivos de saúde, esses novos protocolos tem modelos de

[1] http://openconnectivity.org/

[2] https://www.hl7.org/fhir/

comunicação diferentes, os quais podem levar ao desenvolvimento de novos módulos de avaliação de contexto.

- Distribuição das regras de monitoramento em nuvem. Nessa proposta, as regras de monitoramento podem estar disponíveis em nuvem, e o Gateway BLE apenas a aplicaria. Nesse mesmo contexto, o Gateway BLE pode enviar informações de uso dos dispositivos a serviços em nuvem, os quais podem aplicar modelos de aprendizado de máquina para aferição de novas regras de monitoramento.

- Aplicar modelos de regras de monitoramento a partir de informação de contexto a outras tecnologias de transmissão. Apesar do modelo de controle de fluxo baseado em créditos ser utilizado no BLE, modelos semelhantes podem ser aplicados a outras tecnologias de transmissão.

# Bibliografia


[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. Lee & Seshia, 2011.

[3] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems", *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, 2011.

[4] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "RFC 7252: The constrained application protocol (CoAP)", *Internet Engineering Task Force*, 2014.

[5] D. Locke, "MQ telemetry transport (mqtt) v3. 1 protocol specification", *IBM developerWorks Technical Library], available at http://www. ibm. com/developerworks/webservices/library/ws-mqtt/index. html*, 2010.

[6] J. Nieminen, C. Gomez, M. Isomaki, T. Savolainen, B. Patil, Z. Shelby, M. Xi, and J. Oller, "Networking solutions for connecting bluetooth low energy enabled machines to the internet of things", *Network, IEEE*, vol. 28, no. 6, pp. 83–90, 2014.

[7] T. Savolainen, K. Kerai, F. Berntsen, J. Decuir, R. Heydon, V. Zhodzishsky, and E. Callaway, *Internet Protocol Support Profile*. Bluetooth Special Interest Group - SIG, 2014.

[8] M. Isomaki, J. Nieminen, C. Gomez, Z. Shelby, T. Savolainen, and B. Patil, "IPv6 over BLUETOOTH low energy", 2015.

[9] Bluetooth SIG, "Bluetooth specification version 4.2", *Bluetooth Special Interest Group*, 2014.

[10] D. P. Simons, "Consumer electronics opportunities in remote and home healthcare", *2008 Digest of Technical Papers - International Conference on Consumer Electronics*, pp. 1–2, 2008.

[11] R. Carroll, R. Cnossen, M. Schnell, and D. Simons, "Continua: an interoperable personal healthcare ecosystem", *IEEE Pervasive Computing*, pp. 90–94, 2007.

[12] IEEE, *ISO/IEEE 11073-20601: Health informatics - Point-of-care medical device communication - Part 20601:Optimized exchange protocol Standards*, 2010.

[13] IEEE, *ISO/IEEE 11073-10101: Health informatics - Point-of-care medical device communication - Part 10101:Nomenclature*, 2008.

[14] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology.", *Sensors (Basel, Switzerland)*, vol. 12, no. 9, pp. 11734–53, 2012.

[15] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4", in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pp. 232–237, IEEE, 2012.

[16] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies", *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 80–88, 2010.

[17] M. C. Domingo, "A context-aware service architecture for the integration of body sensor networks and social networks through the ip multimedia subsystem", *IEEE Communications*, no. January, pp. 102–108, 2011.

[18] J. Aragues, A.; Martinez, I.; Del Valle, P.; Munoz, P.; Escayola, J.; Trigo, "Trends in entertainment, home automation and e-health: Toward cross-domain integration", *Communications Magazine, IEEE*, vol. 50, no. 6, pp. 160–167, 2012.

[19] S. Spinsante and E. Gambi, "Remote health monitoring by OSGi technology and digital TV integration", *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1434–1441, 2012.

[20] H. Viswanathan, B. Chen, and D. Pompili, "Research challenges in computation, communication, and context awareness for ubiquitous healthcare", *IEEE Communications Magazine*, vol. 50, no. 5, pp. 92–99, 2012.

[21] M. Bazzani, D. Conzon, A. Scalera, M. Spirito, and C. Trainito, "Enabling the iot paradigm in e-health solutions through the virtus middleware", in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pp. 660–665, 2012.

[22] C. Seeger, K. Van Laerhoven, and A. Buchmann, "MyHealthAssistant: An Event-driven Middleware for Multiple Medical Applications on a Smartphone-mediated Body Sensor Network.", *IEEE journal of biomedical and health informatics*, vol. 2194, no. c, pp. 1–9, 2014.

[23] V. Raychoudhury, J. Cao, M. Kumar, and D. Zhang, "Middleware for pervasive computing: A survey", *Pervasive and Mobile Computing*, vol. 9, no. 2, pp. 177–200, 2013.

[24] F. Palumbo, J. Ullberg, A. Stimec, F. Furfari, L. Karlsson, and S. Coradeschi, "Sensor network infrastructure for a home care monitoring system.", *Sensors (Basel, Switzerland)*, vol. 14, no. 3, pp. 3833–60, 2014.

[25] A. Benharref and M. A. Serhani, "Novel cloud and SOA-based framework for e-health monitoring using wireless biosensors.", *IEEE journal of biomedical and health informatics*, vol. 18, no. 1, pp. 46–55, 2014.

[26] W. Xiaonan, L. Deguang, C. Hongbin, and X. Conghua, "All-IP wireless sensor networks for real-time patient monitoring.", *Journal of biomedical informatics*, 2014.

[27] A. A. Rezaee, M. H. Yaghmaee, and A. M. Rahmani, "Optimized Congestion Management Protocol for Healthcare Wireless Sensor Networks", *Wireless Personal Communications*, vol. 75, no. 1, pp. 11–34, 2013.

[28] A. A. Rezaee, M. H. Yaghmaee, A. M. Rahmani, and A. H. Mohajerzadeh, "HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks", *Journal of Network and Computer Applications*, vol. 37, pp. 216–228, 2014.

[29] D. Niyato, E. Hossain, and S. Camorlinga, "Remote patient monitoring service using heterogeneous wireless access networks: architecture and optimization", *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 412–423, 2009.

[30] G. Zhou, Q. Li, J. Li, Y. Wu, S. Lin, J. Lu, C.-Y. Wan, M. D. Yarvis, and J. A. Stankovic, "Adaptive and Radio-Agnostic QoS for Body Sensor Networks", *ACM Trans. Embed. Comput. Syst.*, vol. 10, no. 4, pp. 48:1–48:34, 2011.

[31] Z. Ren, G. Zhou, A. Pyles, M. Keally, W. Mao, and H. Wang, "BodyT2: Throughput and time delay performance assurance for heterogeneous BSNs", *2011 Proceedings IEEE INFOCOM*, pp. 2750–2758, 2011.

[32] B. Liu and Z. Yan, "CA-MAC: A Hybrid context-aware MAC protocol for wireless body area networks", *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pp. 213–216, 2011.

[33] A. Ludovici and A. Calveras, "A Proxy Design to Leverage the Interconnection of CoAP Wireless Sensor Networks with Web Applications", *Sensors*, vol. 15, no. 1, pp. 1217–1244, 2015.

[34] D. F. de Souza Santos, F. Bublitz, H. Almeida, and A. Perkusich, "Integrating IEEE 11073 and Constrained Application Protocol for Personal Health Devices", in *Proceedings of the 2014 ACM Symposium On Applied Computing*, ACM, 2014.

[35] D. F. de Souza Santos, A. F. Martins, H. Almeida, and A. Perkusich, "UPnP and IEEE 11073: Integrating personal health devices in home networks", in *Proceedings of 11th IEEE Consumer Communications and Networking Conference*, ACM, 2014.

[36] D. F. de Souza Santos, R. Berkoff, C. Stevens, P. Jangwoong, and P. Jeon, "Sensormanagement:1 device for UPnP, version 1.0, standardized DCP (SDCP)", *UPnP Forum Standardized DCPs, available at http://upnp.org/specs/smgt/smgt1/*, 2013.

[37] D. F. de Souza Santos, A. F. Martins, A. F. de Albuquerque Rodrigues, J. L. Do Nascimento, A. Perkusich, and H. O. De Almeida, "Personal Health Data Hub", 2014. US Patent Application - 14/148,548 (USPTO).

[38] D. F. de Souza Santos, A. F. A. Rodrigues, M. F. Pereira, H. O. Almeida, and A. Perkusich, "An Interoperable and Standard-Based End-to-End Remote Patient Monitoring System", in *Encyclopedia of E-Health and Telemedicine*, pp. 260–272, IGI Global, 2016.

[39] D. F. de Souza Santos, A. F. Martins, H. Almeida, and A. Perkusich, "IEEE 11073 and Connected Health: Preparing Personal Health Devices for the Internet", in *Proceedings of ICCE 2014*, IEEE, 2014.

[40] D. F. de Souza Santos, A. Perkusich, and H. Almeida, "Standard-based and distributed health information sharing for mHealth IoT systems", in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 94–98, 2014.

[41] D. F. de Souza Santos, A. Perkusich, and H. Almeida, "A personal connected health system for the internet of things based on the constrained application protocol", *Computers and Electrical Engineering Journal*, 2015.

[42] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals", tech. rep., 2007.

[43] A. Castellani, S. Loreto, A. Rahman, T. Fossati, and E. Dijk, *Best practices for HTTP-CoAP mapping implementation*. IETF work in progress, 2012.

[44] E. Rescorla and N. Modadugu, "Datagram transport layer security version 1.2", *Internet Engineering Task Force - IETF*, 2012.

[45] W. Colitti, K. Steenhaut, and N. De Caro, "Integrating wireless sensor networks with the web", in *Extending the Internet to Low powerand Lossy Networks (IP+ SN 2011)*, 2011.

[46] ITU-T, *H.810 : Interoperability design guidelines for personal health systems*, 2013.

[47] J. G. Rhoads, T. Cooper, K. Fuchs, P. Schluter, and R. P. Zambuto, "Medical device interoperability and the Integrating the Healthcare Enterprise (IHE) initiative", *Biomed Instrum Technol*, pp. 21–7, 2010.

[48] IEEE, *ISO/IEEE 11073-10201: Health informatics - Point-of-care medical device communication - Part 10201:Domain information model*, 2008.

[49] F. Touati, O. Erdene-Ochir, W. Mehmood, A. Hassan, A. B. Mnaouer, B. Gaabab, M. F. A. Rasid, and L. Khriji, "An experimental performance evaluation and compatibility study of the bluetooth low energy based platform for ecg monitoring in wbans", *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[50] W. Bronzi, R. Frank, G. Castignani, and T. Engel, "Bluetooth low energy performance and robustness analysis for inter-vehicular communications", *Ad Hoc Networks*, vol. 37, pp. 76–86, 2016.

[51] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in bluetooth low energy networks", *Communications Letters, IEEE*, vol. 16, no. 9, pp. 1439–1441, 2012.

[52] C. Gomez, I. Demirkol, and J. Paradells, "Modeling the maximum throughput of bluetooth low energy in an error-prone link", *Communications Letters, IEEE*, vol. 15, no. 11, pp. 1187–1189, 2011.

[53] E. Mackensen, M. Lai, and T. M. Wendt, "Performance analysis of an bluetooth low energy sensor system", in *Wireless Systems (IDAACS-SWS), 2012 IEEE 1st International Symposium on*, pp. 62–66, IEEE, 2012.

[54] K. Mikhaylov, "Simulation of network-level performance for bluetooth low energy", in *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pp. 1259–1263, IEEE, 2014.

[55] V. Chawathawortcharoen, V. Visoottiviseth, and R. Takano, "Feasibility evaluation of 6LoWPAN over Bluetooth low energy", *arXiv preprint arXiv:1509.06991*, 2015.

[56] H. Wang, M. Xi, J. Liu, and C. Chen, "Transmitting IPv6 packets over Bluetooth low energy based on BlueZ", in *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pp. 72–77, IEEE, 2013.

[57] T. Savolainen and M. Xi, "IPv6 over Bluetooth low-energy prototype", in *Aalto University Workshop on Wireless Sensor Systems, Aalto, Finland*, 2012.

[58] J. Yim, S. Kim, N.-K. Kim, and Y.-B. Ko, "IPv6 based real-time acoustic data streaming service over Bluetooth low energy", in *Communications, Computers and Signal Processing (PACRIM), 2015 IEEE Pacific Rim Conference on*, pp. 269–273, IEEE, 2015.

[59] H.-s. Kim, J. Lee, and J. W. Jang, "Blemesh: A wireless mesh network protocol for bluetooth low energy devices", in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pp. 558–563, IEEE, 2015.

[60] P. Makris, D. Skoutas, and C. Skianis, "A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration", *Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, 2013.

[61] E. Kartsakli, A. S. Lalos, A. Antonopoulos, S. Tennina, M. D. Renzo, L. Alonso, and C. Verikoukis, "A survey on M2M systems for mHealth: a wireless communications perspective.", *Sensors (Basel, Switzerland)*, vol. 14, no. 10, pp. 18009–52, 2014.

[62] D. P. Tobon, T. H. Falk, and M. Maier, "Context awareness in WBANs: a survey on medical and non-medical applications", *IEEE Wireless Communications*, vol. 20, no. 4, pp. 30–37, 2013.

[63] A. El Mougy, A. Kamoun, M. Ibnkahla, S. Tazi, and K. Drira, "A context and application-aware framework for resource management in dynamic collaborative wireless M2M networks", *Journal of Network and Computer Applications*, vol. 44, pp. 30–45, 2014.

[64] Z. Yan and B. Liu, "A context aware MAC protocol for medical Wireless Body Area Network", *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 2133–2138, 2011.

[65] S. Rezvani and S. A. Ghorashi, "Context aware and channel-based resource allocation for wireless body area networks", *IET Wireless Sensor Systems*, vol. 3, no. 1, pp. 16–25, 2013.

[66] L. Carvalho, R. Campos, and M. Ricardo, "Context-aware low-energy Wi-Fi sensor

networks for e-health", *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, no. Healthcom, pp. 361–365, 2013.

[67] P. Karla, S. I. Saffer, V. P. Gurupur, and S. C. Suh, "Identification of class of services in the Internet and a proposed approach to traffic prioritization at layer 3", *2012 Proceedings of IEEE Southeastcon*, pp. 1–5, 2012.

[68] S. Ullah and K. S. Kwak, "An ultra low-power and traffic-adaptive medium access control protocol for wireless body area network.", *Journal of medical systems*, vol. 36, no. 3, pp. 1021–30, 2012.

[69] S. Cheng, C. Huang, and C. C. Tu, "RACOON: a multiuser QoS design for mobile wireless body area networks.", *Journal of medical systems*, vol. 35, no. 5, pp. 1277–87, 2011.

[70] M. Fengou, G. Mantas, D. Lymberopoulos, N. Komninos, S. Fengos, and N. Lazarou, "A new framework architecture for next generation e-Health services.", *IEEE journal of biomedical and health informatics*, vol. 17, no. 1, pp. 9–18, 2013.

[71] S. Jung and W. Chung, "Wireless machine-to-machine healthcare solution using android mobile devices in global networks", *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1419–1424, 2013.

[72] IEEE, *11073-00101-2008 - Health Informatics - PoC Medical Device Communication - Part 00101: Guidelines for the Use of RF Wireless Technology*, 2008.

[73] K. Hartke, "Observing resources in CoAP - draft-ietf-core-observe-16", *Internet Engineering Task Force - IETF*, 2014.

[74] R. González Gómez, R. D. Hughes, D. Bogia, K. Shingala, L. Kermes, M. Lima, R. Herbster, L. Pfützenreuter, E.and Ott, J. R. Barr, G. Schatz, and R. Strickland, "Personal health devices transcoding white paper, v1. 4", *Bluetooth SIG*, 2012.

[75] R. H. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. M. Behlen, P. V. Biron, and A. S. Shvo, "HL7 clinical document architecture, release 2", *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 30–39, 2006.